3D People Detection in Domestic Environments

Frederik Hegger

Publisher: Dean Prof. Dr. Wolfgang Heiden

University of Applied Sciences Bonn-Rhein-Sieg,
Department of Computer Science

Sankt Augustin, Germany

March 2012

Technical Report 02-2012

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ISSN 1869-5272

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Master Thesis

3D People Detection in Domestic Environments

Frederik Hegger

A thesis submitted to the
Bonn-Rhine-Sieg University of Applied Sciences
for the degree of
Master of Science in Autonomous Systems

Referee and Tutor:	Prof. Dr. Paul G. Plöger
Referee:	Prof. Dr. Gerhard K. Kraetzschmar
Referee:	Dipl.-Inform. Nico Hochgeschwender

Submitted: October 2011

ABSTRACT

The ability of detecting people has become a crucial subtask, especially in robotic systems which aim an application in public or domestic environments. Robots already provide their services e.g. in real home improvement markets and guide people to a desired product[1]. In such a scenario many robot internal tasks would benefit from the knowledge of knowing the number and positions of people in the vicinity. The navigation for example could treat them as *dynamical moving objects* and also predict their next motion directions in order to compute a much safer path. Or the robot could specifically approach customers and offer its services. This requires to detect a person or even a group of people in a reasonable range in front of the robot. Challenges of such a real-world task are e.g. changing lightning conditions, a dynamic environment and different people shapes.

In this thesis a 3D people detection approach based on point cloud data provided by the *Microsoft Kinect* is implemented and integrated on mobile service robot. A Top-Down/Bottom-Up segmentation is applied to increase the systems flexibility and provided the capability to the detect people even if they are partially occluded. A feature set is proposed to detect people in *various pose configurations and motions* using a machine learning technique. The system can detect people up to a distance of 5 meters. The experimental evaluation compared different machine learning techniques and showed that standing people can be detected with a rate of 87.29% and sitting people with 74.94% using a *Random Forest classifier*. Certain objects caused several false detections. To elimante those a verification is proposed which further evaluates the persons shape in the 2D space. The detection component has been implemented as s sequential (frame rate of 10 Hz) and a parallel application (frame rate of 16 Hz). Finally, the component has been embedded into complete *people search task* which explorates the environment, find all people and approach each detected person.

[1] **A robot acting as guide for customers in TOOM home improvement markets -** `http://www.tu-ilmenau.de/fakia/Toomas.6483.0.html?&L=1`

ACKNOWLEDGMENTS

I would like to express my sincere thanks and appreciation to the following persons for their support:

Advisors:

- Prof. Dr. Paul G. Plöger
- Prof. Dr.-Ing. Gerhard K. Kraetzschmar
- Dipl.-Inform. Nico Hochgeschwender

The whole RoboCup@Home b-it-bots team:

- Jan Paulus
- Mike Reckhaus
- Christian Müller
- Thomas Breuer
- Sven Schneider
- Geovanny R. Giorgana Macedo
- Zha Jin
- Jose Antonio Alvarez Ruiz

Others:

- Ana Mijoc
- My family and friends

Thank you all,
Frederik Hegger

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Robotic Systems or AI (Artificial Intelligence) components in general are recently escaping from the laboratories and entering into the real world. Many new cars come along with different drive assistant and control systems which relieve the driver and also increase the safety in daily traffic. Since this all takes place in the real and unconstrained world, such a system has to cope with the high dynamic of it. One well-known and major dynamic of the real world is the human being itself. Humans can move slowly, fast, change the direction suddenly or even stop immediately. For an autonomous car moving in the daily traffic, it would increase the security immense, if the car could detect pedestrians in the vicinity. Through additional tracking and motion prediction, the car could detect if the upcoming situation could cause in an accident or not. If so, the car could adjust the path to avoid a collision with the pedestrian. This could prevent from many accidents between cars and people. In order to react on such situations, an AI systems has to be able to perceive them reliably. An AI system can be a component inside a car, like described or even a service robot at home.

Whereas navigation, object perception and manipulation are crucial capabilities of a service robot, there is still one capability which is almost the most crucial one, namely the *Human-Robot-Interaction* (HRI). A mobile service robot which is not able to interact with its owner would be a useless and expensive toy. And therefore the robot needs to know where actually people are in its surrounding. The people awareness also helps to operate mobile service robots safe in coexistence with humans and react on their movements and actions. Even components like face- or gesture recognition are a kind of dependent on the knowledge of people positions. Usually these components do not work on large distance ranges and require the human to be close to the robot. The people detection can help to find persons in the larger vicinity and then approach them for further actions.

Furthermore, the awareness of people also supports other robot tasks like SLAM (*Simultaneous Localization and Mapping*) or dynamic obstacle avoidance. A local path planner can adjust e.g. a preplanned path according to the dynamical movements of people which might block the actual path. Another popular example for an application field is the surveillance, especially of large cities. In times of increasing terrorism, the governments are induced to find suspects quickly or prevent an assassination very early. In such situations, a people detection component eases the search for possible suspects.

The previously described application fields point to the importance of people detec-

tion over a huge range of different domains. Although, the detection of people in domestic environments has been an active research domain since years, it can be still considered as not solved robustly. There is still a need of reliable and robust algorithms which can handle different conditions, e.g. different lightning conditions, different environments (indoor and outdoor) or crowed scenes. Even the applied sensor type plays a very important roll since different sensor devices come along with different advantages and disadvantages. In the last couple of years there have been a couple of different sensor types which have become more or less standard for people detection:

- Laser Range Finder
- Monocular Cameras
- Stereo Cameras

But there a still various other sensors which could be utilized to detect people like e.g. *Time-of-Flight cameras* (TOF). In comparison to a *3D Laser Range Finder* (LRF), a TOF camera can cover a scene with a single shot in a few milliseconds, while a 3D LRF needs a few seconds to acquire a complete 3D scan (due to the rotational operation mode). The scene might already have been changed strongly, although the complete 3D scan is not yet established. Moreover, 3D LRF's are heavy and large compared to a conventional camera system. Typical examples for TOF cameras are the *MESA Swissranger*[2] or the *PMD[vision] CamCube*[3]. Both cameras provide a high frequency of 25 Hz - 54 Hz and cost between $9,000 and $12,000. The drawback of these cameras is the low resolution of max. 204 x 204 pixels and the very restricted opening angle which makes it hard to detected people in a wide scene. A person might even not be completely visible in the camera due to this reason. Stereo cameras have been also quite popular in robotic tasks, but they provide only a sparse 3D depth map, because the correspondence problem [30] can not be solved for each pixel in the image.

Recently, another 3D camera has become available as commercial product - so called *RGB-D cameras*. These cameras provide a RGB-image and a 3D point cloud at the same time. To determine the depth, a infrared pattern is projected into the scene and captured by a camera. Based on the deformation of the captured infrared pattern the distance information can be calculated. In comparison to the stereo camera, the RGB-D cameras provide an almost complete representation of a scene as point cloud. Only for special (shiny) surfaces like a monitor screen or glasses, the depth information can not be determined. Currently only one RGB-D cameras is commercially available - the *Microsoft Kinect*[4]. The camera has become very popular in research, since it is low-cost

[2]**MESA Swissranger** - `http://www.mesa-imaging.ch/`
[3]**PMD[vision] CamCube** - `http://www.pmdtec.com/`
[4]**Microsoft Kinect** - `http://www.xbox.com/kinect`

and operates with a high frequency of 30 Hz. It provides a resolution of 640 x 480 pixels in both - color and depth image. Furthermore, it comes along with a wide opening angle and its far distance range (up to 10 meters). Hence, a large part of the environment can be covered with a single shot.

In our previous work [19], we developed a people detection system based on two LRF. Two LRF's were mounted in two different heights - one in leg height and one in waist height. The detection was applied separately in each layer and the resulting positive detection were fused together in order to further strengthen the reliability of the overall system. The major drawback of the proposed approach in our previous work has been the less information provided by the two LRF. The world is captured only in two laser scan slices at two different levels. If a person is only visible in one layer, e.g. occluded by a cupboard, then the reliability decrease rapidly. Hence, in this thesis a new people detection technique is proposed which exploits 3D point cloud data. The final system is integrated on our mobile service robot *Care-O-bot 3* [16] which participates regularly at the RoboCup@Home[5] competitions.

The remainder of this thesis is structured as follows: in Section 2, the current state-of-the-art in the domain of people detection is surveyed. Section 3 describes the general problems of detecting people in domestic environments. The proposed approach is presented in Section 6 in which all subcomponents like the preprocessing, segmentation and classification are explained in detail. The final system has been tested and evaluated in a standalone fashion and fully integrated on a real mobile service robot. The results are described and illustrated in Section 7. Finally, the thesis is concluded in Section 8 and a future outlook is given in Section 9.

[5] **RoboCup@Home** - `http://www.robocupathome.org`

Chapter 2

RELATED WORK

During several years of research and development in robotics, several sensors have been become popular for a wide range of application fields. For the detection of people, current state-of-the-art approaches have mostly used sensors like the LRF, monocular cameras or stereo cameras. In general, each sensor type comes along with different advantages and disadvantages. In our previous work [18], the field of people detection and tracking has been survey in detail. Therefore, this Section summarizes briefly the current state-of-art and recent achievements in this domain.

2.1 Laser Range Finder

Approaches which are based on LRF have to cope with less information, since a LRF perceives the current scene only in a single (usually horizontal) scan line. On the other hand it provides a high operating frequency, accuracy and long distance range (up to 30 meters). A common technique to detect people in laser scans is the deployment of a supervised machine learning approach. A LRF is mounted in leg height and the captured scans are segmented into smaller clusters according to a so called *Jump Distance Criteria* [31]. For each cluster a set of simple geometric features [1] is calculated like e.g. the width, circularity and linearity. An AdaBoost or SVM (Support Vector Machine) machine learning technique is used to train a generic model offline and afterwards determining online weather a current segments belongs to a human or not. A major disadvantages is the single view point to the scene. If a person is standing behind e.g. a cupboard such an approach would not be able to detect this person.

In [9] and [26] one or two additional LRF have been added to the system architecture in order to gain a more sophisticated view to the scene. With these extensions the waist or/and the head can be covered additionally to the legs. These additional view points enable the handling of partial occlusions of persons by objects like cupboards, tables or small shelfs. A shape model and probabilistic voting is applied to fuse the information of the different layers into a more robust detection hypothesis. The shape model of both approaches assumes a static model and is only dedicated to standing persons. Both approaches are not able to find sitting persons, because for each layer a separate model is trained, dedicated to a specific body part. If a person is sitting, the correlation of a human body part (waist, head) and the respective LRF layer changes and is not taken into account by the proposed approaches.

2.2 Vision

People detection approaches based on vision have applied standard CCD (Charge-coupled Device) cameras [4, 21, 3] and stereo cameras [39, 28] as primary sensor. Approaches using the stereo camera have usually not taken the depth information into account for the detection it self. The detection has been done in the 2D image and the 3D information were only be used to determine the 3D position of a detected person in the image. The major target application of monocular camera approaches has been pedestrian tracking and the surveillance of buildings, rooms or large cities. For such tasks, e.g. to monitor an office room, a camera is mounted statically somewhere at the ceiling. Due to this kind of setup the background usually stays the same or only changes slightly. Hence, background subtraction has been a popular technique [46, 14] to reduce the search space in an image. But considering a mobile service robot as a target platform such approaches are not applicable at all.

For the preliminary detection of people in images, the face is one prominent feature. In [4, 21] a classifier based on *Haar-like features* [45] is applied to find the persons face in the image. Hu *et al.* [21] use the initial face detector to find the upper body (through an rigid shape model). Once the upper body is found a color model of that region is used for further tracking. In [4] two trackers are maintained at the same time. The first tracker is initialized with the face detection and tracks then the Haar-like features in the subsequent images. But the face is usually not always visible to the camera. Therefore a second instance keeps track of the upper body in the same manner as for the face tracking. Although, there has been a detection phase in the beginning, the detection is actually performed through the tracking. This principle is called *detection-by-tracking*.

In comparison to monocular cameras, stereo cameras provide another dimension - namely depth information. Munoz-Salinas *et al.* [28] have utilized color and gradient information to segment the scene regarding to similar color regions. A face detector is applied to find an initial prediction where a human could be located. A 3D body has been introduced which consists of two ellipses, the head and the upper body. Since the position and the size of the face is known through the face detection, they assume that the size of the body-ellipse must be placed below the head and always two times bigger than the fitted head ellipse. A color model and the depth information are forwarded to a particle filter which tracks the person overtime. Another approach by Satake and Miura [39] established a set of predefined depth-templates for the upper body of a person. In total three templates were been created: one from the persons front, back and side. A template matching technique then tries to find a person in the image according to these templates. Other approaches have applied well-known techniques like *Implicit Shape Models* [43], *Haar-like-features* [48] or *Histograms of Oriented Gradients* [42]. But all these approaches operate only in 2D space.

2.3 3D-Sensors

The domain of people detection in three dimensional space has been only discovered sparsely until now. However, Spinello *et al.* [41] proposed an approach to detected persons in a single 3D point cloud produced by a rotating LRF. The point cloud is subdivided into a fixed number of virtual 2D horizontal laser scans in different heights. Afterwards, they a apply the approach from [1] for each layer with an adopted and extended feature set. For each layer a separate model is trained. Previous approaches have manually established a shape model to fuse the information of multiple layers. Spinello *et al.* automatically learn the displacements of the different segments to each other. The major advantage of their approach is that the proposed approach is able to handle partial occlusions, since if a few body parts are occluded, there are still some parts left which are visible. But the presented approach has one major disadvantage: for a specific virtual scan, one classifier is trained respective to the expected body part in this height. Hence, this approach assumes that the persons must be standing.

In [2], they proposed an approach which detects people in stereo vision data. Therefore, the 3D input data is projected to a 2.5D polar perspective map. The resulting grid map is then segmented using cell statistics. Finally a set of different shape- (e.g. moments) and geometric features (e.g. width, height, volume) is calculated and serves as input for a machine learning classifier. The presented approach shows good results at outdoor scenes where the landscape is flat. But in indoor scenarios, like in an apartment this kind of segmentation would probably not result in reasonable clusters since many objects are close to each other. This would then also effect the distribution of the feature space and probably result in a decreasing detection rate.

A similar method is described in [29]. The actual scene segmentation is quite simple and based on the assumption that the position of the ground-plane is known. They extract a 3D slice from the original data with all points above the ground and only a maximum height. The maximum height is adaptive and adjusted according to the mean and standard deviation of all points of the previous scan. The remaining points are projected into a 2D representation and segmented into single clusters. Each cluster is then tracked over time. For each established 2D cluster, they recover then the respective 3D points. By performing a Principle Component Analysis (PCA), the two principle planes of the human body are obtained. A normalized histogram of the first two principle planes is calculated and fed to a classifier (SVM). While tracking the clusters, they also compute a motion score which is established based on the information of the object's size, traveled distance and variations in size and velocity. In a second classification stage the motion score and and the outcome of the first classification are fed into a second SVM. Although the final detection rate is promising, the proposed approach detects people only in upright positions.

Spinello and Arras [40] so far is the only approach considering people detection

with a RGB-D sensor. Their approach is a combination of detections in 3D depth and 2D image data. The visual detection is based on a *Histogram of Oriented Gradients* (HOG). Inspired from the HOG detector [13], they propose a novel HOD-descriptor for the detection in 3D data, which is composed of a *Histogram of Oriented Depths* (HOD). Both histograms are used as input feature vector for an SVM-based machine learning classifier. Fusing both information - visual HOG and HOD - together, yield in a robust people detection system. The ability to perform the detection in real-time (30 Hz) was accomplished through GPU (Graphics Processing Unit) implementation of the proposed technique. Their evaluation compared their approach against several other (depth- and visual-based) on the same dataset. They state that the combination of HOD and HOG outperforms all other approaches. But the results show only a comparison to their previous work.

Beside the presented literature, the *OpenNI*[6] project developed a software framework for natural interfaces. These framework is especially dedicated to the new RGB-D cameras like the Microsoft Kinect or the ASUS Xtion PRO LIVE[7]. The project aims to get away from traditional interface like the mouse or keyboard and go further and use the whole human body as an extended interface. Therefore the framework provides a skeleton tracker which tracks the major joint configuration of the human body. Embedded in this functionality, a people detection and tracking module tries to find people in a maximum range of 3.5 meters. Unfortunately, the applied algorithms are closed-source and it can only be assumed which techniques they have used. But it seems, that the people detection applies some kind of adapting background subtraction. During several experimental test runs of the system, the people detection works well if the camera is static and not moving. But when moving the camera by hand or even on a moving robot, many false positive detections can be observed. Even when the skeleton for a detected person is initialized (requires a specific initialization arm posture) the tracking can not be executed robustly when the camera is moving.

[6] **OpenNI -** `http://www.openni.org`
[7] **ASUS Xtion PRO LIVE -** `http://us.estore.asus.com/index.php?l=product_detail&p=4001`

Chapter 3

PROBLEM STATEMENT

The previous chapter described several detection approaches based on a variety of different sensor types. This proves that the detection of people is not yet a solved problem and that there is no general solution how to tackle this problem.

One fundamental characteristic of people is the variation of their appearances. In context of visual information, the variations relate to the skin color, hair style or their current clothes. These human properties have a large deviation through the whole mankind. This make it challenging to generalize over these properties in order to find a common description for a person. On the other hand, there is the human shape which on the first glance seems to be more general than the visual information. But also in shape the human undergoes various changes. Although the structure of a human body looks always similar, there are still the height and the width which might vary a lot. Humans have also many degrees of freedom (DOF) which allows them to build up different poses and moving in different speeds. They could sit on a chair, be bend to pick up a bucket or just lying on a bed. Further, people are not always moving in free space and so they are not all the time visible completely. Their body can be partially occluded when a person e.g. is sitting behind a table or just standing behind a cupboard. Then the lower body part might not be visible to the sensor at all.

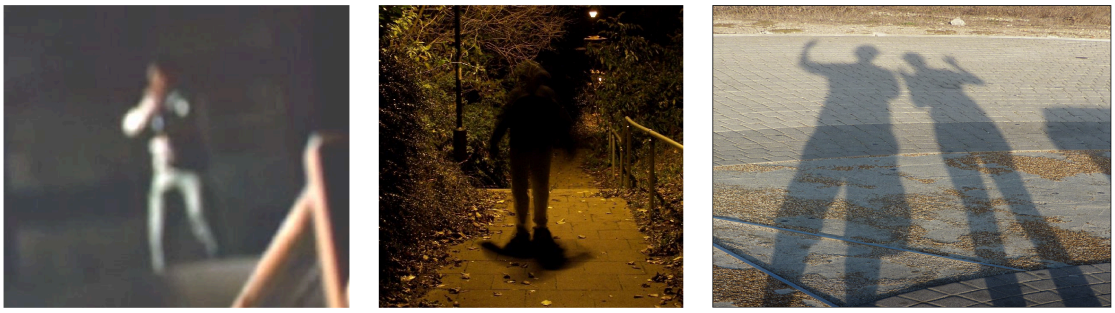
Despite from the described human body properties above, the applied sensor(s) play(s) an additional important role. Table 3.1 illustrated a brief summary of the major advantages and disadvantages of available sensor types. When using a camera and the respective color information, there are several challenges which can occur during runtime. When the camera is not mounted in a static manner, but rather on a mobile service robot, motion blur caused by the robots movement is a very big challenge. And if the robot is not moving, the person definitely will do. Also the lightning conditions might change when moving between different rooms or even buildings. And what happens if there would be a electricity blackout? Figure 3.1 illustrates some example images of such typical vision problems. In 3.1(c) two persons are not directly visible in the image, but throw two shadows onto the ground. How many people would a visual people detection system find in such a case? And at which position? A human would infer that the persons are close to the camera, but an algorithms would probably estimate the persons position within the center of the image. Such simple examples can already be a huge challenge for a vision system.

	Advantages	**Disadvantages**
LRF	- high frequency - large range - high accuracy	- only 2D data - single scan line
Monocular camera	- color information - high frequency - "infinite" range	- dependent on illumination - motion blur
Stereo camera	- see monocular - 3D data	- see monocular - sparse 3D data
TOF camera	- dense 3D data - high precision	- no color information - restricted range - low resolution - moderate frequency
Kinect camera	- color + depth data - large range - high frequency	- noisy 3D data - no depth data for special surfaces - not applicable outdoors

Table 3.1: Sensor characteristics

In comparison to the vision, a LRF provides far lesser information. In our previous work [19], two LRF's were mounted in different heights in order to increase the field of view (FOV). Only one LRF in leg height causes many false positive detections, because there are many small and leg-like objects (e.g. chair-legs) in this specific height. And at large distances a leg is only represented by a few points (<5) in laser scan. This kind of setup has been used in many state-of-the-art approaches, but due two the lack of information (only 2D laser scan slices), such systems are highly susceptible when a human is partially occluded.

In 3D data, the detection becomes even more challenging. Point clouds generated by a TOF- or Kinect camera can easily consists of > 100.000 data points. Due to this huge amount of data, the actual preprocessing and segmentation of the scene plays a very important role. Only with a intelligent solution it is possible to guarantee an acceptable frame rate (which of course depends on the final application field). Even the segmentation of 3D data itself is an own research domain.



(a) Motion blur (b) Poor illumination (c) Shadows

Figure 3.1: Typical vision problems

Chapter 4

REQUIREMENT ANALYSIS

As proposed, the final system aims an application in domestic environments. For robotic applications in such environments, the RoboCup@Home competition has been become an nearly standardized test platform. A rulebook specifies the appearance of the environment and defines several tests-cases, each dedicated to one or several capabilities (e.g. manipulation, perception or HRI).

4.1 Use-Cases

The current version of the rulebook [36] already specifies three tests which include the detection of people. These tests are *FollowMe*, *WhoIsWho* and *EnhancedWhoIsWho*. In *FollowMe* a specific person is going to be tracked and followed over a certain distance. To start the following, initially a person needs to be detected which the robot should follow. In both *WhoIsWho* tests, the scenario is a party in an apartment and the robot should find and identify people (*WhoIsWho*) and for *EnhancedWhoIsWho* bring those people a pre-ordered drink. From those test specifications, a least two use cases can be derived which will be described in the following paragraphs:

Use-Case 1: Finding all people in an apartment

An undefined number of people is gathering together in an apartment to have a party. A service robot is acting as a kind of a butler and should e.g. welcome new guests and bring their coats to the coatroom. Additionally, the robot can ask the guests if they want to have an additional drink or snack. For this purpose, the robot needs to explore the environment (maybe based on a set of predefined routes or landmarks) and continuously check whether it can detect a person in the current camera frame or not. If a person is found, the robot has to determine the position and approach the person. It is not further specified where and in which posture the people might occur.

Use-Case 2: Find a calling person

A service robot might not be driving around all the time (like in use case 1) and annoying e.g. the owner by asking all time questions. Therefore it might be more efficient if the robot is standing somewhere and is willing to take orders. When a guest is calling the robot, e.g. from behind, it could turn towards the voice source

(i.e. through an additional sound localization component like in [22]) and try to find the person who has called the robot to come. A solution would be to take the closest detected person w.r.t the detected sound angle. The robot could then approach the person and ask for the purpose of calling.

4.2 Resulting Requirements

When designing and developing new functionality, there is the wish of high generalization and optimal results for a huge range of different conditions. However, considering a given problem realistically, there are always some restrictions. This holds also through for the people detection. But first we will outline the major requirements which have to be fulfilled by the system. The following requirements for the people detection application result from the problem statement in the previous Section and the described use cases in the above paragraphs.

Person independent:

Properties like skin color, hair color or clothes coloring of a person should be neglected by the detection mechanism.

Environment independent:

The environmental structure or objects inside a particular scenario can be anything related to a real home-like apartment.

Markerless detection:

There are no additional markers required like e.g. for motion captured systems.

Non-static camera:

The camera does not have to be mounted in a static manner. It is allowed to move the camera, but the distance and orientation respective to the ground plane must be known.

Various lightning conditions:

The detection must be robust against changing lightning conditions, since they might vary a lot when driving around with a robot.

Various postures:

People do not always stand or sit in the same way. The component must be to a certain degree robust against those variations in posture.

Chapter 5

DESIGN

The following chapter describes the design process on both, hardware- and software level. The final system has been integrated and tested in several scenarios a on real mobile service robot. The following paragraphs will present the applied hardware briefly and explain the software integration into an already existing software framework.

5.1 Hardware Setup

The available robot platform is a *Care-O-bot 3* (COB3) robot [16] (see figure 5.1) which has been developed by the Fraunhofer IPA[8]. It is equipped with several sensors and actuators. For navigation tasks, the COB3 owns a omnidirectional base which allows to move in any direction. Two SICK S300 LRF (front and back) and one Hokuyo URG-04LX on top of the mobile base are used for map-building, localization and obstacle avoidance. The trunk has in total 4 DOF to tilt and pan the entire body. On the back side the COB3 accommodates a 7 DOF KUKA LBR arm and a 7 DOF Schunk gripper to manipulate objects. Grasped objects can be placed on a tray which can be moved up and down which allows the robot to carry more than one object at the same time. The actual head is flippable to look forward and backward. A microphone is mounted for speech recognition purpose in order to give commands to the robot. To perceive the environment, the head consists of two AVT Pike 145 monocular cameras which are combined to a stereo camera. Recently, the original TOF camera has been replaced through a Microsoft Kinect camera.

The comprehensive hardware of the COB3 already provides the necessary requirements for the proposed people detection approach. The required sensor, the Microsoft Kinect in the head is mounted in a height of ≈1.45 meters. This yield in large coverage of the actual scene and an adequate FOV to detect person in various poses. A even larger FOV can be obtained by using the DOFs of the trunk and the head. Since the detection mechanism has not only been tested in a static manner, but also in more complex scenarios (e.g. "find all people in the apartment"), where other components like e.g. the base were required.

[8] **Fraunhofer IPA** - `http://www.ipa.fraunhofer.de`

Figure 5.1: Jenny - A Care-O-bot 3 robot platform

5.2 Software Framework

A basic software framework for the COB3 has been already provided by the Fraunhofer IPA. This includes basic navigation, arm movements and controlling all the specific hardware parts. The underlying software framework is based on ROS [32] - the Robot Operation System. The modular structure of ROS eases the reusability of components and enhances the development process. It provides advanced visualization tools for various types of data, a large range of low level driver for standard components and connections to other software frameworks. Other tools like *rosbag* are extremely helpful to evaluate a specific component or algorithm. Parts of the presented experiments in Section 7 made use of this tool to evaluate the system with different parameters but on exactly the same dataset.

Beside the Fraunhofer code base, the b-it-bots team has been developing, additional low and high level packages during their preparation and participation at recent RoboCup@Home competitions. A reasonable part of components has been ported from the old hardware platform (VolksBot "Johnny" [7]) to the new COB3 platform.

5.3 Integration Aspects

The existing software components, so called packages, provide functionalities for a large scale of applications. They are designed in a modular manner in order to be very flexible in connecting different small components to a much higher level task. Figure 5.2 illustrates some example packages of the b-it-bots framework and their dependencies to others.

The people detection method itself is established as an own component package (see yellow box in figure 5.2). The interface of the component (see figure 5.3) is very clearly. As

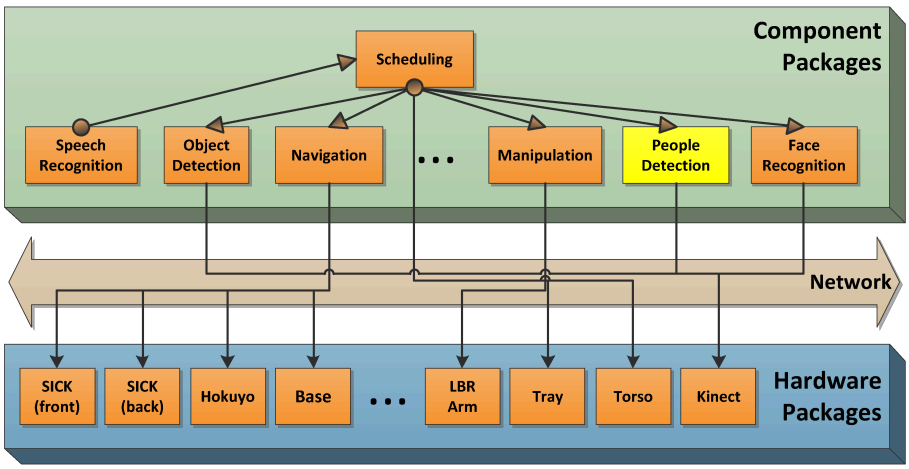


Figure 5.2: Overview of the existing software configurations and the related components

it input, it requires data from the Kinect (point cloud + RGB information) and coordinate transformation information. These informations are provided as topics and the component must only subscribe to the specific topics in order to perceive the desired data. For control purposes, two services are advertised by the people detection component: a start and a stop service. In large scale systems with many components running at the same time, it is crucial to pause those functionalities which are currently not needed. Especially when the computational effort of a few functionalities is very high, this is an important point to be considered. After the launch of the component, it is in pause mode and runs only in about 1 Hz which is sufficient to wait for incoming service requests. With a call of the start service, the people detection will be performed at full frequency. The detection results will be published as a ROS message including the following information for each detected person:

- Real 3D position
- 3D position with a relative safety distance
- Height

The latter two items can be used in scenarios where the robot must search and navigate to people in order to recognize and remember them. Therefore the robot approaches the person up to a predefined safety distance. The height information can be used to adjust the tilt angle of the camera head to have the persons face ideally in the center of the RGB image for the face recognition. Of course, the people detection component could provide more information (e.g. the point cloud region, width or depth), but current demands do not require those additions.

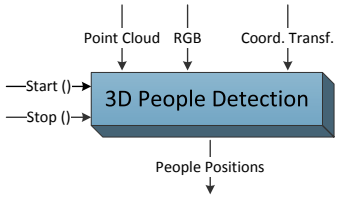


Figure 5.3: Interface of the people detection component

5.4 Component Design

The previous paragraphs illustrated the overall software environment and already defined a minimal interface for the desired people detection functionality. In this Section, an overview of the class structure is given. The problem of detection people involves many intermediate steps. In order to maintain the reusability of the software and make it understandable for others, the source code has been split into different classes and functions according to their dedicated functionality. The complete class diagram is shown in figure 5.4. The main class (*PeopleDetection3D*) builds the main instance of the detection algorithm. As result of the function call of *getPersons()*, a list of detected person is returned. The segmentation of the point cloud is realized as a separate class, since it might be useful for other future components. It is designed in such a way, that it can extended with additional 3D segmentation strategies. All machine learning techniques have been implemented according to a common interface which eases replacement of and extension by other machine learners. Small functionalities like e.g. distance or conversion functions have been grouped into different categories and provided in a common ROS package. This promotes the reusability of code and bugs can be found much faster and reliable when code is reused for other components in the system.

The people detection mechanism is encapsulated into a single ROS node according to the previously defined interface of the component. The node subscribes to the required Kinect topics and publishes the detection results on another topic. In this configuration the processing of the data is sequential. For a multi-core system, the parallelization of the processing pipeline would be interesting to increase the frame rate. Several frameworks, like e.g. *OpenMP*[9], provide the capability to parallelize several parts of the code. But also ROS provides something similar through the distributed system architecture. Since each node is started as an own process, the costly computation tasks can be encapsulated into several separate nodes and will result in a more or less parallelized processing pipeline. On a multi-core computer with 4 or 8 physical cores, this staged processing can result in 57% higher frame rate compared to a pure sequential processing. Compared to an OpenMP or even more to a GPU-based implementation, this workaround can be realized with less effort and does not need any special knowledge. The distribution of low-level functionality

[9] **OpenMP** - `htpp://www.openmp.org`

(like e.g. the surface normal compuation or subampling) has another adavantage in larger systems. The existing software components of the b-it-bots includes components which also need the prioir compuation of the surface normals or the subsampling like e.g. the object detection component [27]. In cases where the whole functionaility of the object and people detection is relaized in each node, the computation of certain subtasks is done twice. But in a setup where e.g. the surface normal computation is outsourced into a separate node, the processing is just done once and saves costly computation time.

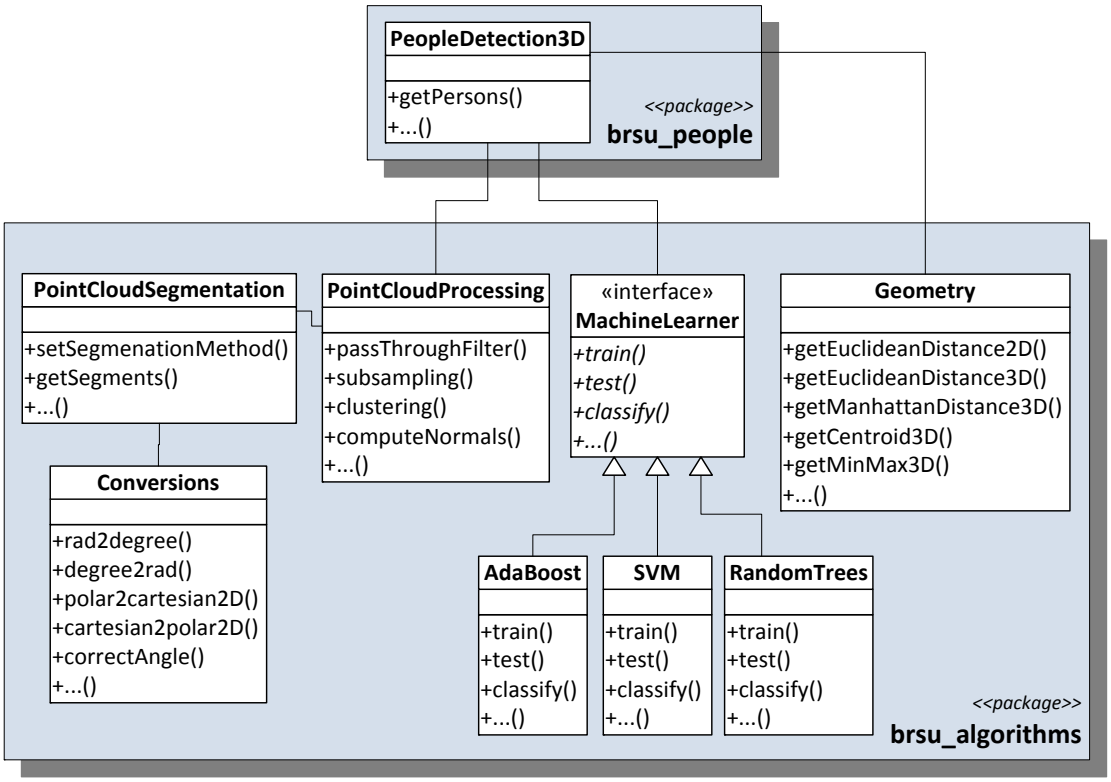


Figure 5.4: UML diagram of the people detection software structure

Chapter 6

APPROACH

In this chapter the developed 3D people detection approach and its subcomponent are described in detail. The following paragraphs are structured in adaption to the decomposed subtasks of the developed 3D people detection approach, namely a preprocessing-, a segmentation- and a detection stage. Additionally, this chapters outlines the main contribution of our approach and describes which assumptions have been made.

6.1 Overview

The 3D data has been acquired with a Microsoft Kinect camera which provides a synchronized data structure of RGB values and a 3D point cloud. The raw point cloud is very large and consists over 300.000 points. Due to performance purposes, the point cloud is initially cropped down to a region-of-interest (ROI) and subsampled. Afterwards, the remaining data is split into smaller clusters using a layered sub-division of the scene and a top-down/bottom-up segmentation technique. A machine learning classifier is applied to label the resulting 3D clusters either as human or non-human based on a established feature set. An overview of the complete processing pipeline is depicted in figure 6.1.

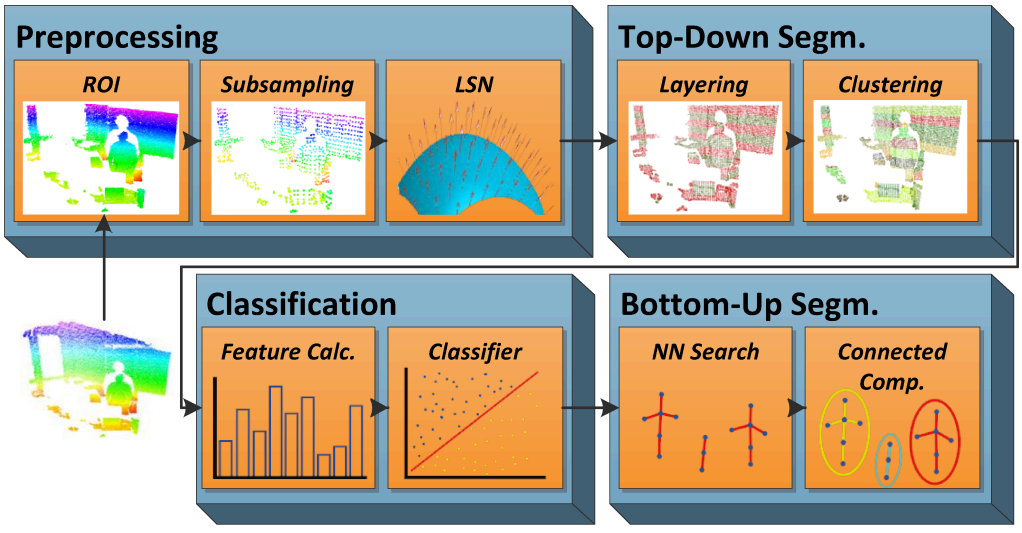


Figure 6.1: Overview of the people detection processing pipeline

6.1.1 Contribution

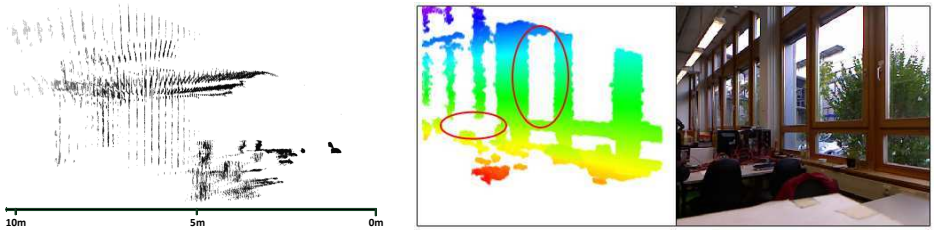
Current literature [40, 39] have proposed approaches which mainly target the detection of people in upright postures. For surveillance purpose, e.g. pedestrian tracking this assumption might hold true, but in domestic environments the people can take several postures like e.g. sitting on a chair. In such environments people are even likely to be occluded by other objects. A person can sit on a chair behind a table or is even standing behind a cupboard. Then a large part of the human body is not visible at all. The main contribution of this thesis is a new feature descriptor based on local surface normals and the capability to detect persons in various poses/motions and even if they are partially occluded like sitting behind a table or desk.

A further contribution is the applied segmentation scheme. Standard 3D segmentation techniques based on graph theory [44] or region-growing [33] need several seconds to compute a set of clusters of an unstructured point cloud. In order to fulfill the objective of a frame rate of at least 1 Hz, a naive segmentation strategy is proposed which is fast and yield in the necessary performance for the detection of people. Further, the proposed approach is independent of the environment and people can also be detect if the camera is moving, i.e. no background subtraction (like [47, 34, 10]) is applied. This enables the application on moving platforms and increases the range of applications fields. Although, the detection procedure is designed to be as general as possible, there are still a few limitations which will be explained in the next paragraph.

6.1.2 Assumptions

The Microsoft Kinect cameras has been become very popular when it comes to 3D perception. Its low price and the high frame rate are one reason for the high impact of the sensor. Despite the fact that the Kinect provides some good properties, it also comes along with a few disadvantages concerning the accuracy. In [23], the authors have investigated these issue and have concentrated on the increasing depth error at larger distances. In their results, they showed that the depth error can accumulate up to 4 cm at a maximum distance of 5 meters. In close range (up to 1 meter) the actual error is in average in the millimeter range. Above a distance range of 5 meters, the depth discretization error results in visible depth slices 6.2(a). At 5 meters the distance between those slices amounts about 5 cm and at 10 meters the distance between the slices can accumulate up to 35 cm.

Referring to this result, the maximum allowed distance range is restricted to 5 meters which is also sufficient for domestic environments. Beside the depth limitation, also the maximum perceivable height has been restricted. People are assumed to appear usually in a certain height above ground. In [24] a comprehensive survey has been made along the human height distribution of men and women of different age. The maximum evaluated average height has been 189 cm. When adding the additional standard deviation of 2.75



(a) Slicing effect (b) Missing data

Figure 6.2: Illustration of two drawbacks of the Microsoft Kinect camera: (a) the depth error increases with the distance and results in a slicing effect. (b) Through IR absorbing or shiny surfaces the data exhibits blobs where the depth can not be determined.

cm, it ends up in a maximum height if 191.75 cm. For simplicity, these has been round up to 2 meters and taken as maximum height for the perception. Concluding the latter restrictions, the following ROI is defined:

- **Depth:** $0m > x < 5m$
- **Width:** $-\infty > y < +\infty$
- **Height:** $0m > z < 2m$

All those points which do not fulfill these definitions will be rejected and not further considered. This cropping increases as well the frame rate of the overall system, since a less amount of points need to be processed.

Regarding the Kinect as sensory input device, it must be said, that the application of this sensor is limited to indoor applications. In outdoor scenarios the sun has a huge impact, because it is a natural IR-source. Even in the shadow below of a balcony, the Kinect was not able to produce a representable point cloud. Only small and still undistorted chunks of point were received.

The last assumption concerns about the detectable postures of a human. According to the proposed feature vector in Section 6.4.2, a reasonable part of the upper body must be visible. This means, poses like lying on a bed or floor will not be detected. All other poses like sitting on or behind objects, standing in various poses (even on one foot) or running around will be detected through the proposed people detection approach. But when people are standing to close together (distance between two person <10 cm), they are considered as one common object due to the applied segmentation technique.

6.2 Preprocessing

During each run, initially an unstructured point cloud from the Kinect is acquired. Each raw point cloud consists of exactly 307.200 points due to the resolution of 640 x 480

pixels of the camera.

Invalid Points

As mentioned in the problem formulation (Section 3), in some cases the depth can not be determined. Those points are kept in the data structure and marked as *NaN*'s (Not a Number). Through a further investigation on the amount of NaN's w.r.t. the original amount of data, it could be observed that the average number of *NaN*'s in a full point cloud (307.200 points) is around 23.98% (73.667 points). Of course, this result is based on the environment in which the data was taken. In this case, the data (1137 samples) was captured while the COB3 was navigating autonomously accross an apartment-like environment. But removing these *NaN*'s already reduces the amount of data significantly. In order to save expensive computation time this removal of *NaN*'s is done simultaneously with the building of the *ROI*.

Region-of-Interest

The exact dimension of the ROI has already been described and justified in the previous Section (see 6.1.2). For throwing points away which to not fulfill the ROI description, the *PCL-library* [38] provides a so called *passthrough filter*. It is axes-based and can only be performed on one axis at same time. But the defined ROI has restrictions for two axes. If using the PCL passthrough functionality, the filter had to be run two times to all the points. Therefore, an adopted filter is implemented which loops only once through the whole data set. Each point is being checked if it is not a *NaN* and fulfills the defined ROI restrictions ($0m > depth < 5m$ and $0m > height < 2m$). This small change saves 1/3 of computation time compared to the standardized PCL filter implementation.

Subsampling

The remaining points inside the ROI are further reduced by a subsampling routine to make the point cloud more sparse. Further other components like a object recognition, it is important to have a high point density because the object are in general very small. In comparison to the proposed system in this thesis, the actual target object -the human - is much larger. Hence, a such high initial point density is not needed. With an increasing distance it decreases anyway due to the slicing effect. In order to have an equally distributed point density a long the maximum distance range of 5 meters and to further reduce the amount of data to a minimum, the subsampling is applied. For each input cloud a 3D grid with predefined cell size is overlayed over the full point cloud. The points inside each box are merged (averaging over all three axes) to a single new point. So, the higher the cell size is, the sparser the point cloud gets. In the final system setup a cell size of *3 cm* x *3 cm* x *3 cm* has been utilized which still maintains the desired accuracy

for the local surface normal estimation and simultaneously reduces the total amount of data points further. A pleasant side effect of the subsampling is the implicit smoothing of noise since the impact of outliers is reduced due to the averaging over all points inside a cell.

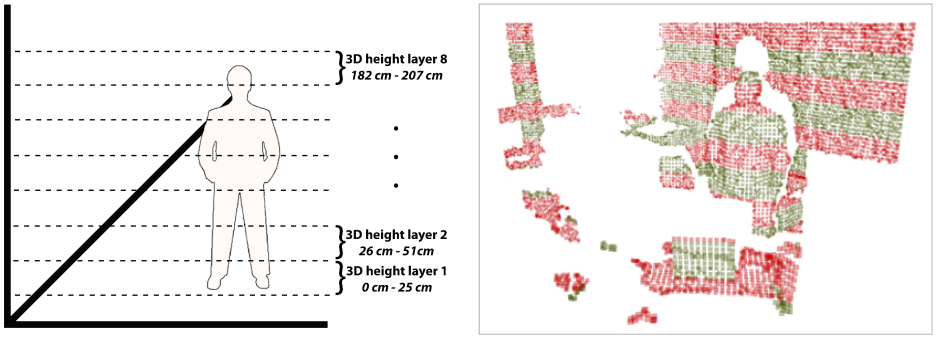
Local Surface Normals

For the latter classification (Section 6.4.2) a new feature descriptor is composed which is mainly based on local surface information, namely local surface normals [25]. A surface normal is estimated by fitting a plane to the k-nearest neighbors of the target point. A more detailed description of the applied algorithm is presented in [37]. The computation of the surface normals is scheduled before the actual segmentation because it results in a more accurate estimation of the particular surface normal. If the normals would be calculate after the segmentation for each particular cluster, the accuracy for those points which lie on the border of a cluster would be significant lower, since a reasonable part of the neighborhood might already belong to another cluster. Computing the surface normals for all points is one of the computational expensive tasks in the people detection processing pipeline. It benefits from the previous rejection of points and can therefore be performed in roughly 30 ms (on an Intel i7 2.7 GHz 8-core processor). In comparison to that, a normal estimation for the raw input cloud would take about several seconds dependent on the size of the chosen neighborhood.

These preliminary steps (excluding the normal computation) already reduces the point cloud down to a minimum. They are necessary to keep the processing cost as low as possible and therefore a higher frame rate can be achieved.

6.3 Segmentation

The segmentation of large 3D point clouds is a costly and complex task, since the crucial decision where to start or end a cluster is heavily dependent on the desired accuracy of the final application. Unfortunately, current 3D segmentation approaches like region growing or graph-based approaches have the drawback of huge computational complexity and consume a high amount of costly computation time. Another demand to the segmentation is the ability to find people even if they are occluded by an other object or even sitting on a table. Taking the latter example, a region growing approach would merge the points of the person and the table into one cluster. Then a classification of this cluster would be a real challenge to still find the person. Therefore, a two staged *top-down segmentation* technique (see figure 6.3) is proposed, whose general idea is to first partition the initial point cloud into a fixed set of different 3D height layers and then start to segment the each layer separately into smaller clusters.



(a) Layering principle (b) Layering result

Figure 6.3: In the layering stage, each point cloud is divided into a set of 3D layers according to a manually defined slice height. In the final system, a height threshold of 25 cm has been applied. The different colored points in (b) indicate the different height layers.

6.3.1 Layered Subdivision

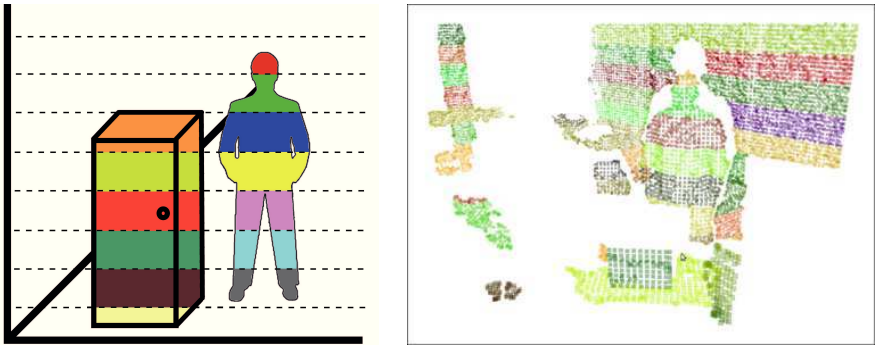
An acquired point cloud has been so far preprocessed and shrinked to an amount of points, so that the segmentation can be performed in an affordable time. In the next step, the point cloud is divided into several height layers. Different from [41], the height layers do not consist of virtual 2D laser scan slices, but rather of complete 3D layers. Figure 6.3 illustrates the principle of the 3D layering (in 6.3(a)) and an example of the resulting layers of a real scene (in 6.3(b)). The applied algorithm can be explained as follows:

Let $\mathbf{P} = \{p_1, ..., p_N\}$ be a point cloud with $p_i = \{x, y, z\}$ and N equal to the number of points in the point cloud. Then $\mathbf{P}$ is split into a fixed number of 3D layers $\mathbf{L} = \{l_1, ..., l_M\}$ with

$$M = \frac{|Z_{max} - Z_{min}|}{SH}$$

where Z_{min} and Z_{max} are the minimum and maximum height values of the predefined ROI (min = 0.0m, max = 2.0m) and SH is the desired slice height. Then, for each layer l_j the minimum and maximum height (of points which they should contain) is being calculated. Assuming a predefined slice height of *25 cm*, then the first layer l_1 contains only points with $0.0m >= p_i(z) <= 0.25m$. The remaining layers $l_2, ...l_M$ will be established according to this principle. Respective to the experimental evaluation, a slice height of *25 cm* has been considered as optimal threshold.

The characteristic of this 3D layering is, that a person (and of course the whole scene) is split into several height parts. This helps especially in situation where a person might be partially and physically connected to other objects in the environment.



(a) Schematic segmentation (b) Segmentation result

Figure 6.4: Each layer is segmented into clusters using a *Euclidean Clustering* approach. The different colored points in (b) indicate the segmented 3D clusters.

6.3.2 Euclidean Clustering

Now that the point cloud is decomposed into several 3D height layers, the actual segmentation generates for each layer l_j a sequence of small clusters $\mathbf{C} = \{c_1, ..., c_O\}$ where each cluster $c_{j,k}$ contains a subset of points located in l_k. The segmentation applies a *Euclidean Clustering* technique [37], whose advantage is, that it is less parameterized than other approaches like *k-means* [17] or *mean-shift clustering* [11].

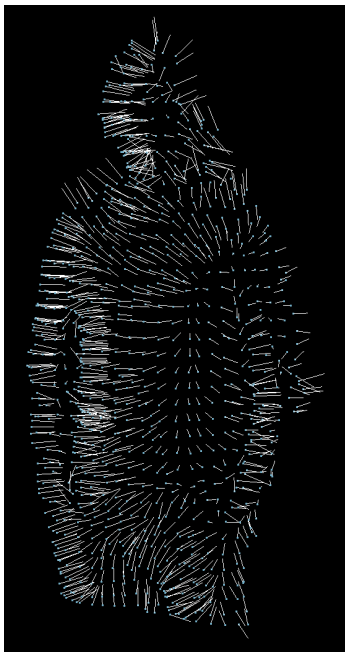
Only a distance threshold $thres_{EuclDist}$ has to be defined, which says that a target point is only added to the current cluster, if the Euclidean distance is smaller than the specified threshold. But $thres_{EuclDist}$ also determines weather there are many small clusters ($thres_{EuclDist} \leftarrow 0$) or only a few large clusters ($thres_{EuclDist} \rightarrow \infty$). As mentioned before, a grid-size of *3 cm* for the subsampling has been used. According to this dimensions and a certain amount of noise, the threshold has been set to $thres_{EuclDist} = 2 \times grid\_size$ in order to ensure that two persons which stand close to each other are not merged to a single cluster.

This kind of more fine-grained clustering has the advantage over a clustering *without* prior layering when one object is partially connected to another object. If e.g. a person is sitting on a table, the presented approach creates several smaller clusters for both objects. Instead, the pure Euclidean Clustering will end up in a single cluster which consists of the table and the person, since the person is sitting very close to the table or has put the arms on it. The user-defined slice height plays also an important role for the performance of the segmentation. A reasonable small height ends up in really tiny clusters whose information (i.e. local surface normals) are not sufficient for a robust classification. On the other hand, a large slice height creates also large clusters (where two or more objects would get merged to a single cluster) which would abolish the specific advantage of the proposed segmentation stage.

6.4 Detection

The segmentation stage, described in the previous Section, produces a sequence of 3D clusters. The task of the classification stage is now to assign a label to each segmented cluster, so either *human* or *non-human*. In [20], the authors detect planes by clustering local surface normals according to their vector orientation and take only those clusters with nearly horizontal or vertical normal orientation. Inspired from this approach, we composed a new feature vector to describe a 3D cluster for the detection of people.

6.4.1 Feature Descriptor



In many state-of-the-art approaches [1, 26, 41], geometric and statistical features (like width, height, number of points, etc.) have been very popular. Hence all these features are dedicated to the classification of 2D clusters, we propose a new kind of feature vector which is more dedicated to three dimensional datasets, namely a *histogram of local surface normals*. If considering a regular apartment, such an environment tend to consists usually of walls, tables/desk, shelfs, chairs and various other smaller objects. Thus, a reasonable part consists of horizontal and vertical planes. Whereas the human body has a more cylindrical appearance. Exactly this property can be expressed with the estimated local surface normals. The distribution over those normals is stored as a fix-sized histogram. Since the applied machine learning approaches utilizes only a one dimensional feature vector, the three dimensional normal information have to be converted to 1D. Thus, for each normal dimension a separate histogram is created. Beside those histograms, a set of additional other features (adopted from 2D features of [1]) has been added to the final feature vector. Summarizing the explanations above, the overall feature vector is composed of the following information:

- **Local surface normal histogram**: for each dimension of the surface normal a single 1D histogram overall points within a 3D cluster is establish.
- **Width:** $\sqrt{min(p_i(y))^2 - max(p_i(y))^2}$
- **Depth:** $\sqrt{min(p_i(x))^2 - max(p_i(x))^2}$
- **Number of points**: $n = |c_{j,k}|$, where l is the layer in which the cluster k is located in.
- **Distance:** distance to the centroid of $c_{j,k}$.

The additional features like the width and depth of a cluster have been added as a feature, which aims the further reduction of the false positive rate.

6.4.2 Classification of 3D Clusters

Till this stage of the people detection processing pipeline, the actual scene is preprocessed, segmented into small cluster and a feature descriptor is computed for each of those clusters. Based on this feature set, a supervised machine learning technique is applied, whose task is to determine whether the respective cluster belongs to a human or not. In an initial training phase a set of positive and negative samples is presented to the machine learner in order to build up a generic model for the presented input data. The samples used for this training were collected in different environments using a semi-automatic procedure which is described in Section 6.4.3.

Supervised machine learning approaches have been popular through a wide range of application fields, especially in cases of a two-class classification problem. The range of application fields span over the classification of objects, rooms, faces or gestures. And even for the domain of people detection, supervised machine learning techniques like AdaBoost or SVM have been applied quite frequently. AdaBoost-based approaches [26, 1] have been applied to the detection in 2D laser scans while SVMs are more related to the detection in images (e.g. [13]). In order to assure to have the best fitting machine learning techniques applied to the proposed people detection problem, in total three approaches were considered, namely *AdaBoost* [15], *SVM* [12] and *Random Forests* [6]. All three techniques have a more or less common interface. Each techniques requires a fix-sized feature vector as input data and outputs the according label as a single character. They only differ slightly in the parametrization. By defining a common interface (see UML diagram 5.4 on page 23) for all three learners, the performance can be easily compared against each other. In the presented experimental evaluation, the results of the machine learning techniques were compared with exactly the same training and testing sets. As implementations the following framework are utilized:

- **AdaBoost:** Computer vision library OpenCV[10]
- **SVM:** libSVM library[11]
- **Random Forests:** Computer vision library OpenCV[12]

Several experiments with different sample datasets proved that the Random Forest classifier outperforms both other approaches. Thus, the trained Random Forest model

[10] **OpenCV library -** `http://opencv.willowgarage.com`
[11] **libSVM library -** `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
[12] **OpenCV library -** `http://opencv.willowgarage.com`

has been applied as the classifier in the final system setup. The final classifier was trained with 2100 samples.

6.4.3 Acquisition of Training Samples

For the training of the Random Forest classifier, a set of training samples needed to be acquired. The collection of positive and negative training samples can get a unpleasant task, especially when many samples (> 1000) are required for the training and the annotation of each sample has to be done manually. Therefore, a procedure has been integrated to capture both, positive and negative training samples in a semi-automatic manner. Negative samples have been collected with a mobile service robot. Therefore, a map of a part of the university building has been established which at least consisted of an office/laboratory, long corridor and an apartment. For each room a navigation goal has been manually annotated. Then, an automatic procedure generated a random order, in which the rooms should be visited. The robot started navigating autonomously through all the environments and simultaneously segmented each incoming point cloud and labels each extracted cluster as a negative sample. For the whole run it was ensured, that there was no person in the FOV of the robot. This process guarantees that the samples are indeed collected in a random manner. The positive samples were collected with a static mounted Kinect camera. The camera was placed into a laboratory with frequent traffic of people. We then defined a ROI which does not contain any object at all and consequently provides an empty point cloud. If then a person passes the ROI, the segmentation stage extracted the related clusters and labeled them as positive samples.

6.4.4 Graph-based Bottom-Up Segmentation

As final output from the classification stage, a sequence of human classified 3D clusters is being obtained. These "part-based" detections need to be assembled and associated to the respective person. Therefore, a graph-based representation based on the cluster's center (vertices) is created. This has the advantage, that not the whole data points of a cluster needed to be processed and keeps the computational effort lower. Each vertex is then connected to its two nearest neighbors as long as the Euclidean distance between those points does not exceed a certain threshold. Since, each cluster has always the same maximum height (equal to the predefined slice height), the threshold can be derived from this prior knowledge, because the center points of two neighboring clusters can only have a maximum distance of $2 \times slice\_height$. When all points in the cue have been processed, the overall graph can be split in to its *connected components*, which finally build the actual person detection. Due to false positive detections when classifying the extracted 3D clusters, a successful person detection is considered only, if at least three clusters belong to one person.

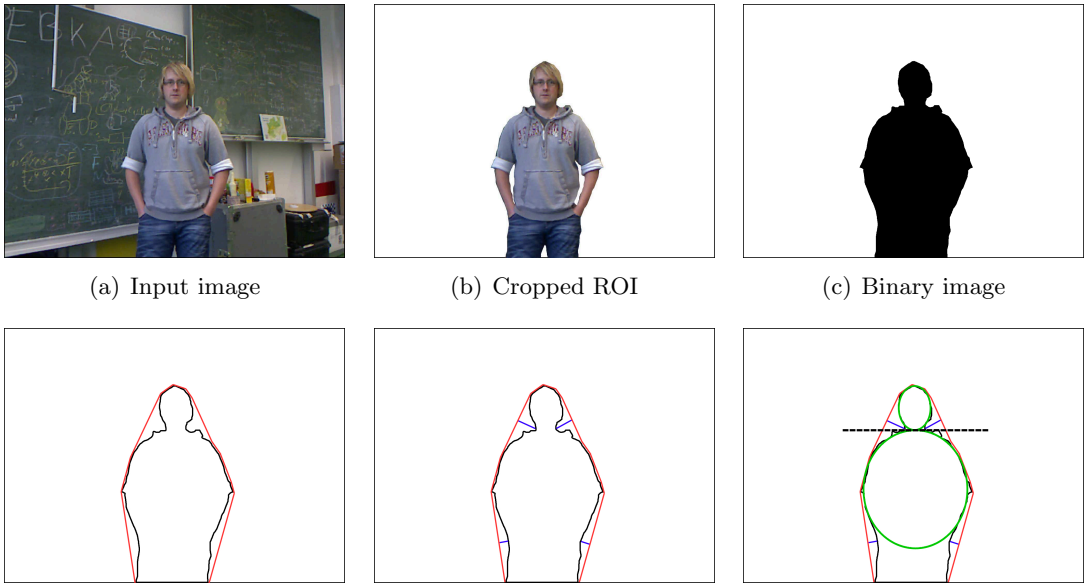
6.4.5 Verification in 2D Space

The previously described detection mechanism is purely based on the 3D point cloud information of the Kinect camera. But one further property of the camera is that it provides additional color information with the RGB image. Since, the experiments in Section 7.2.4 have revealed a high false positive rate, a further verification of the detected people in RGB space is being considered to reduce this rate. Unfortunately, this step has been implemented only partially and thus the verification was *not* part of the experimental evaluation. Therefore only the theoretical thoughts and steps are explained which in our opinion would decrease the false positive detection rate through the application of additional color information.

The general idea of the verification is an evaluation of the persons shape in the 2D space. We consider the human body as shape which is constructed out of three parts: the head, the upper body and the legs. For the verification only the head and the upper body is considered, because this are the parts which are mostly visible even when the person is close to the camera. These parts are extracted and separated using common computer vision algorithms (described in the next paragraph). When both parts are extracted, two ellipses are fitted to the head and the upper body region. Mathematical properties of the ellipses are used as simple weak classifiers and train a final strong classifier using a common machine learning approach. This already explains the general verification principle, but of course many intermediate steps are involved.

After the detection in the 3D point cloud, the points of the persons 3D cluster are transformed to the 2D image space (pixels with x and y). Therefore a nice nice overlay of the point cloud and the respective RGB data is needed. Thus, the camera needs to be calibrated once. ROS provides a technical tutorial[13] and corresponding source code to calibrate the external transformation between the RGB and the IR camera. Once the calibration is done, the region of each detected person in the point cloud can be cut out exactly in the RGB image (fig. 6.5(b)) and converted to a binary image (fig. 6.5(c)). This binary image is then used to calculated the edges (using an Canny edge detector [8]) and get the contour of the person (black line in fig. 6.5(d)). In order to separate the head from the upper body and segment them into two clusters, we make use of the convexity defects (fig. 6.5(e)) of the convex hull (red line in fig. 6.5(d)). When following the contour from the head down to the chest, there area of the head gets thinner towards the neck and then is getting bigger when reaching the shoulder part. This critical point can be found through the convexity defects and mark the area where the head ends and the upper body starts. For each resulting cluster an ellipse is fitted to the remaining contour (fig. 6.5(f)). The human body is now approximated and represented by two 2D ellipses. In a final step

[13] **Kinect calibration tutorial from ROS -** `http://www.ros.org/wiki/kinect_calibration/technical`



(a) Input image (b) Cropped ROI (c) Binary image

(d) Contour + convex hull (e) Convexity defects (f) Fitted ellipses

Figure 6.5: Processing pipeline of the RGB verification

the following mathematical properties of both ellipses are computed:

- **Width**: length of the horizontal ellipse axis
- **Height**: length of the vertical ellipse axis
- **Distance**: the actual distance to the detected person (in the point cloud)
- **Fitting error**: the error between all points of the contour and the fitted ellipse
- **Width ratio**: aspect ratio between the width of the head and the upper body ellipse
- **Height ratio:**: aspect ratio between the height of the head and the upper body ellipse

These properties build the feature input vector to a machine learning approach. The distance feature has been selected since, the size of the ellipses is dependent on the actual distance to camera. To represented the relationship between both ellipses, the ratios for the width and height between both ellipses are considered as further features.

With the proposed verification, we aim a reduction of the false positive rate caused by the detections in the point cloud. Although, this algorithms has not been implemented yet completely, our expectations are quite high that the performance can be improved through this step.

6.5 Drive & Search Behavior

The described people detection techniques as a standalone component is a kind of restricted in its application field. It can be used e.g. to react on passing persons and then each time execute a specific action like taking a photo. But the fact that the proposed people detection approach has been integrated on a real mobile service robot with many actuators, enables for more advanced behaviors. As a showcase and for a high level evaluation (see Section 7.2.7), the people detection has been embedded into a complete people search scenario.

The state machine in figure 6.6 describes the procedure on how to coordinate the "exploration" of an environment, simultaneous search for people and the approaching of found persons. The depicted state machine is implement using SMACH [5], a task-oriented state machine framework. Although it has been developed independently to ROS, it is well connected to the framework and allows to build up dedicated task state machines in less time. The people search behavior was implemented using the Python API of the SMACH framework.

The procedure of driving around and approaching found persons can be described as follows (the related state names of the graph in figure 6.6 are mentioned in the brackets): given a map of the environment and a set of predefined room poses, the robot starts to process the list of poses (*approach_pose_selection*) and navigate to them (*approach_pose_without_retry_non_blocking*). During the navigation, the people detection is constantly running with maximum frequency (*check_if_persons_are_present*). If the first pose (e.g. kitchen) is reached without finding any person the next pose to approach is selected and executed (*get_next_pose*). If a person is detected while approaching a specific room pose, the actual path execution is stopped. If only one person is found, the robot directly starts to approach the found person (*approach_person*). But if multiple people are detected at once a separate state (*select_person_to_approach*) randomly selects a person to approach. In order to do not crash into a person when approaching it, a safety position is calculated already when detecting the respective person. The pose is calculated relative to the robot by subtracting a reasonable safety distance (in the applied scenario 1 meter) from the actual detected position and is then transformed from a relative pose to a world pose in the map coordinate frame. Once the person is approached, the position of this person is saved to a list (*store_person_position*) which maintains the already visited person positions. This avoids from visiting a person several times and get stuck in the exploration since once a person is in the field of view, it would be detected and approached again and again. Additionally, the robot adjusts its camera head according to the height of the detected person and keeps the face in the center of the camera image. If a person is e.g. sitting, the head movement is not sufficient to get the persons face into image center. Therefore the torso is being used to look further down. This functionality is especially

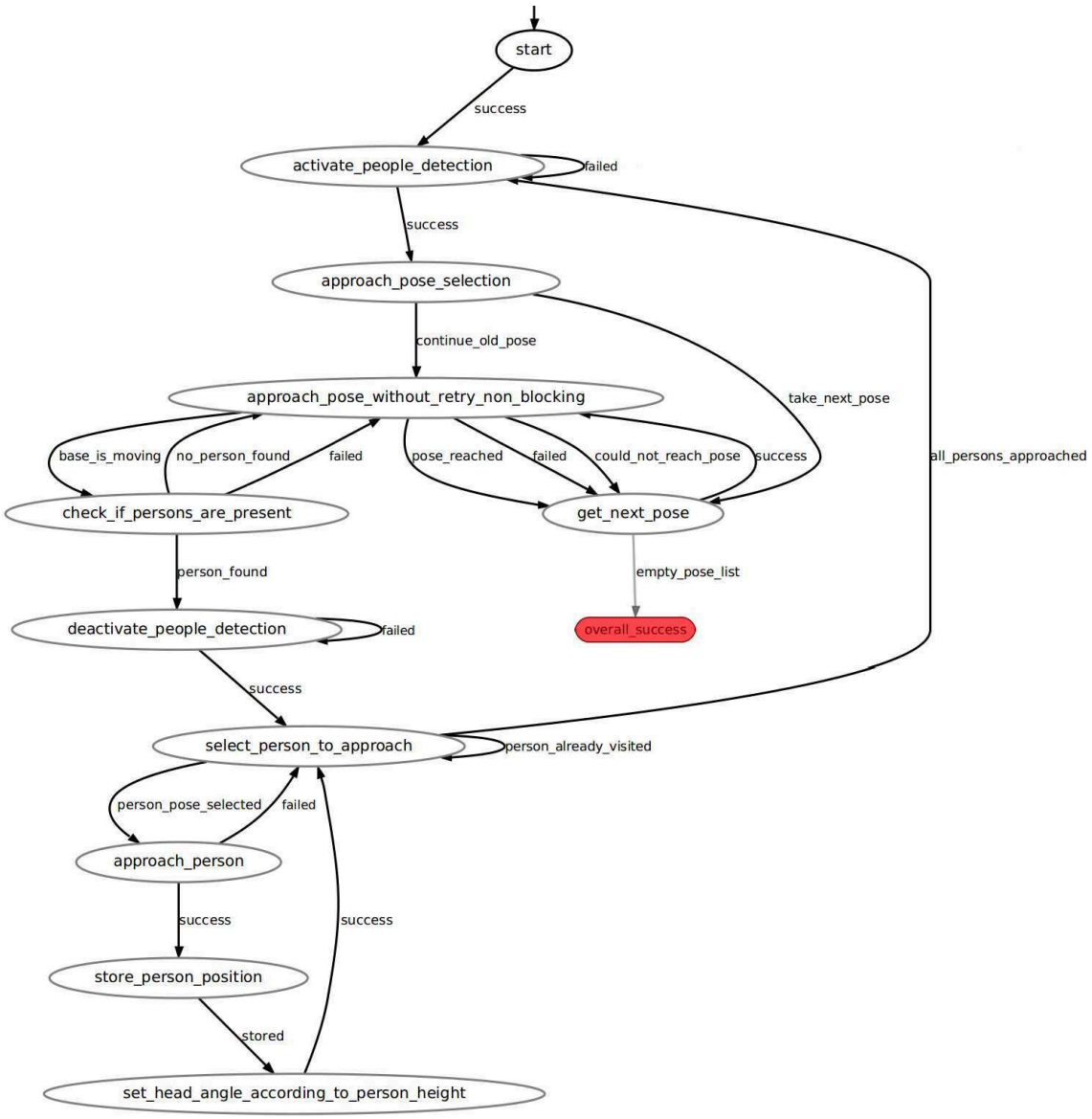


Figure 6.6: State machine of the people search behavior

helpful in scenarios where people do not only have to be found but also to be identified by a face recognition component. After this procedure, it is checked if there are still persons in the list which need to be approach (*select_person_to_approach*). If not, the robot starts either to approach the last pose or if already near that last pose, it will get the next pose (*get_next_pose*). Before actually starting to navigate, the people detection component is activated again.

The presented people search behavior can integrated as a sub state machine into a even more higher level scenario like e.g. in a RoboCup@Home test scenario. In this thesis, the behavior has been used as an additional experiment to evaluate the overall ability to

find people in a domestic environment.

Chapter 7

EXPERIMENTAL EVALUATION

The previous chapter basically described the theoretical as well as the practical aspects of the proposed people detection approach and its integration into a comprehensive robotic system. In this Section, several low-level and even more abstract experiments evaluate the performance of the overall system and its subcomponents.

7.1 General Test Setup

Nowadays, many standard dataset databases (e.g. for face- or object recognition) are available in the Internet, against the performance of a new approach can be tested. This eases the comparison between different approaches a lot. Unfortunately, the domain of 3D people detection does not have such a database yet, because the field is quite new. Through the recent impact of the Kinect camera and already a few existing 3D people detection approaches such a database might be available in the near future. But since there is no public available database yet, an appropriate test environment and test candidates need to be specified.

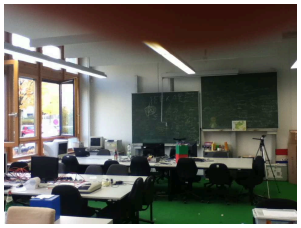
7.1.1 The Environment

The RoboCup@Home laboratory and a few nearby facilities served as primary environments for the evaluation of the people detection system. The laboratory is split into an apartment-like environment (fig. 7.1(a) and 7.1(b)) and a working area for students (fig. 7.1(c)). The apartment consists of a kitchen, two living rooms, and a dining room. Almost everything what a usual apartment consists of. The working area is structured more like an office environment with many chairs and tables. From the working area and the apartment two doors lead to a long corridor (fig. 7.1(d)).

All the presented parts of the test environment(s) have different structures and appearances. The corridor mainly consists of large and long walls with some pillars in between. Compared to this the working/office area is definitely more cluttered. There are many small table legs, chairs, shelfs, cupboards, tables and a set of various small items lying on them like Screens, Keyboard, etc. Finally, the apartment is more or less a combination of both. It is surrounded by a set of walls and consists of two tables, a kitchen with items on the worktop, shelfs and additional things like e.g. a couch. This variety of the environmental structure eases the acquisition of very different training data. For each of the three environments, a set of 200 point clouds and RGB images have been

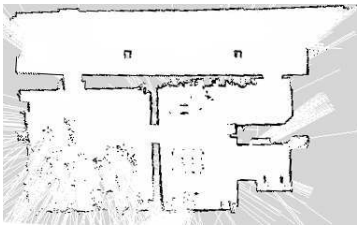
(a) Kitchen

(b) Living room



(c) Working area

(d) Corridor



(e) Map

Figure 7.1: RoboCup@Home laboratory of the Bonn-Rhine-Sieg University of Applied Science

collected randomly without being a person present. This data is especially dedicated to test the system against false positive detections.

7.1.2 The Person Candidates

Considering the shape appearance of people (since this is like they appear in a 3D point cloud), they can have various heights or widths and can undergo different shape transformation. In order to cover these varieties, data was collect from different sized people walking or standing around. As stated in Section 6.4.3 this was done automatically by placing a camera at a place with frequent people traffic. Beside this setup, data was also collect from a fixed group of persons. The persons were told to move in various speeds and were allowed to take any posture they liked to.

7.2 Experiments

The following experiments cover different aspects of the proposed people detection component. The actual structure is based on a bottom-to-top principle. In the lowest level the experiments evaluate specific subcomponents and parameters of the system. In a higher level the system was tested in a black-box manner where the results of standing and sitting persons are compared. Again one step further, the actual computational effort under different setups is considered. Finally at a very high-level stage, the component was undergone an experiment according to a task-oriented scenario.

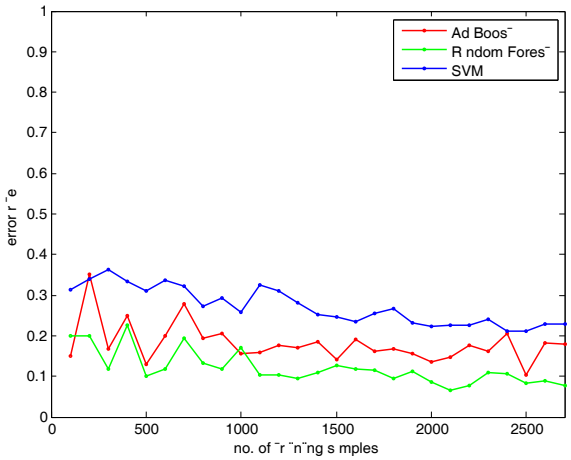
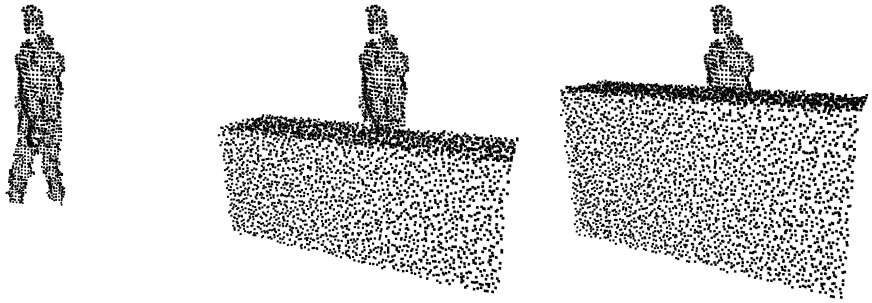


Figure 7.2: Comparison of popular machine learning techniques based on different training sets (using 10-fold cross-validation).

7.2.1 Machine Learning

Description: The decision whether a cluster belongs to a human or not is realized with a machine learning technique. In total three popular machine learners were compared against each other, namely AdaBoost, SVM and Random Forests. The expected outcome of this experiment is that ideally one of the machine learning techniques outperforms both others. Therefore several test sets with different number of samples have been created. Afterwards, each machine learner is trained with all the created sample sets. For the evaluation of the error a k-fold cross-validation was applied. The k was set to 10 iterations, which is a commonly used value in the literature [35]. The experiment was repeated 10 times for each technique and the error was averaged.

Results: The accomplished experiment yielded in a clear tendency towards one machine learning technique. Figure 7.2 depicts the detailed error rates for each specific techniques dependent on the number of samples used for the training. The graph shows that the Random Forest classifier outperforms both other techniques in 92.59% of the applied sample sets with an average error rate of 12.02%. Only for two sample sets the AdaBoost classifier could perform better than the Random Forest and result in an average error rate of 18.06%. The SVM classifier performed worst with a large distance to both other classifiers with an an average error rate of 27.21%. According to this comprehensive classification results, the Random Forest classifier has been chosen as classifier in the final system configuration.



(a) No occlusion (b) 50% occlusion (c) 70% occlusion

Figure 7.3: Different amount of occlusion was added to the input data.

7.2.2 Segmentation

Description: The presented segmentation approach is based on separating the point cloud into several fix-sized layers. The number of layers is dependent on the chosen slice height. This experiment investigated the impact of the predefined slice height to the resulting classification error. The experiment was executed several times with different slice heights ranging from 10 cm to 100 cm. The minimum range value is set to 10 cm according to the investigation that applying a slice height below this value resulted in very few points. This less amount of information is not sufficient to represent a comprehensive distribution. The maximum value (= 100 cm) is set to the half of the maximum perceivable height, thus one requirement for the people detection approach is the ability to detect if they are partially occluded. In each experiment the slice height is constantly increased by 5 cm (when starting at the minimum). As in the first experiment again the 10-fold cross-validation was applied. In order to evaluate the segmentation behavior against occlusions, synthetic generated occlusions (in this experiment it was a kind of a cupboard) was added to the data. The experiment was repeated three times with different amount of occlusion, namely no occlusion (fig. 7.3(a)), 50% (fig. 7.3(b)) and 70% (fig. 7.3(c)) occlusion of the total person. Gaussian noise was added to the synthetic data in order to achieve approximation to the Kinect data.

Results: The diagram in figure 7.4 depicts the cross-validation error w.r.t. the actual applied slice height. If applying the segmentation with no occlusion of the actual person (*green line*), the classification decreases with an increasing slice height till a slice height of approximately 30 cm to 40 cm. Above 50 cm the error converges to an error rate of $\approx 15\%$. In comparision to that, occlusions (*blue + red line*) cause a major boost of the error rate when the applied slice height gets large. Considering the example images in 7.3, the segmentation with a high slice height creates clusters which might consists of both, i.e. parts of the human body and parts of the actual

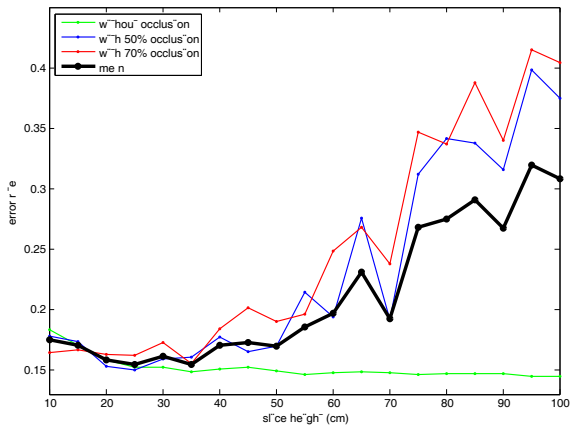


Figure 7.4: Resulting classification errors for various slice heights and amount of occlusions.

object which cause the occlusions, since there is not distance between both objects. With an increasing slice height, the amount of human body and simultaneously object points increases as well inside a single cluster. This disarranges the distribution of surface normals in such a way, that the classifier is not able to build a good model for this data and causes in high error rates. By calculating the mean curvature (*black line*) for all three error curves (*green, blue, red*) and identifying the global minima, the actual applied slice height of 25 cm was determined which yielded in the minimum averaged error of 15.49%. The drop of the error rate (in the range from 10 cm to 20 cm) is caused by the increasing number of point per cluster. A small slice height cause in really small clusters. The amount of the points is then not sufficient enough to form a stable distribution of the surface normals and therefore result in a higher error rate.

7.2.3 People Detection Performance

Description: The following experiment measured the detection rates (DR) under different circumstances. First of all, we defined two categories, namely poses and motions. For the pose category, we evaluated the DR for standing persons, for persons sitting on a chair and for person which are partially occluded (at least 30% of the body). And for the motion category, we evaluate three different motions: not moving, random walking and random running. Due to simplicity, the test were executed in different environments. The performance for random walking person were executed next to the entrance of the Bonn-Rhine-Sieg University of Applied Science where many people enter and leave the building. All the other test were executed either in the RoboCup@Home laboratory or in a real german household environment and with a fixed group of ten people. The particular test procedures (TP) looked as

follows:

1. **Pose → standing:** the persons were asked to position theirselves in a various random positions and usual body postures.
2. **Pose → sitting:** the persons were asked to sit down on a chair and position theirselves in various random positions and usual sitting postures.
3. **Pose → partially occluded:** the persons were asked to move behind a cupboard of 80 cm height up and down in a natural way.
4. **Motion → not moving:** it is identical to the test for standing person and only mentioned for completeness.
5. **Motion → random walking:** the test was execute at the entrance of the Bonn-Rhine-Sieg University. Many people were entering and leaving the building. Even sometimes in small groups.
6. **Motion → random running:** the persons were asked to run in a jogging manner through the FOV of the camera in various paths

For each of the ten persons and the corresponding posture/motion 200 frames have been evaluated. To avoid manual annotation (true positive or false negative detections) of each frame a simplified change detection was applied. Initially the point cloud size (after ROI building) of ten subsequent frame is averaged and stored. In the evaluation phase the size of the recent acquired point cloud is compared to the stored size. If the difference is above certain threshold, the person has entered the cameras FOV. This simplified evaluation was applied for the TC 2, 3 and 6. In case of the TC 1/4, we waited until the person reached a new position and then evaluated each time five frames. For TC 5, each frame had to be manually annotated since the number of persons in the FOV was varying between one and five during the whole test.

Results: The black box evaluation of our system showed a quite robust performance at least for standing person (see table 7.1). The performance is *independent* from the actual distance to the person and is only limited by the predefined maximum distance of 5 meters. But, we observed a degrading detection rate when the person is sitting, e.g. on a chair. The detection rate is significant lower, namely 74.84%. This is due to the fact that the training was only done with standing persons and therefore only the head and the upper body can be detected. The horizontal leg parts can not be detected. If the Random Forest would have trained also with sitting person, there would be clusters whose normal distribution would be similar to horizontal planes (because the upper leg is now parallel aligned). Of course, this would cause in a very high false positive rate. But, when a person is sitting, the

Poses	Detection Rate	Motions	Detection Rate
standing	87.29%	not moving	87.29%
sitting	74.94%	rnd. walk	86.32%
part. occl.	82.35%	rnd. run	86.71%

Table 7.1: Detection rates for different human poses and motions.

upper body is still visible and sufficient for a quite robust detection with the model trained only with standing persons. Although, it is significant lower than detecting standing persons. Persons which were partially occluded, e.g. behind a table or a cupboard, can be detected similar robust to standing person, since only a minority of the lower body is occluded. When the occlusion is so large that the number of visible person clusters is lower than the chosen cluster threshold (evaluated in the next Section), a detection is not possible anymore. For the different motion speeds, only slightly different results could be observed. It does not matter in which speed the person is moving or even standing still, since the detection is done frame by frame. Only the pose configuration is different for the different motions. Figure 7.5 depicts a subset of different configuration which can be successfully detected. The experiment showed that people are detected invarious pose configurations and speeds with a average detection rate of 83.12%.

7.2.4 False Positive Detections

Description: In the previous experimental setups the camera was kept static at a certain position to ease the evaluation. In such a scenario the background only change slightly due to occlusions by crossing persons. In order to evaluate the amount of false positive detections further, this experiment was execute on our COB3 service robot while it was navigating through different environments. The robot moved autonomously across the apartment, the corridor and the laboratory. Simultaneously, the total number of segmented clusters and the respective false positive detections

(a) Bending (b) Random pose (c) Partially occluded + sitting

Figure 7.5: Detections for various pose configurations

were collected. As stated in Section 6.4.4, the number of clusters per person has to be above a certain threshold. The experiment evaluated the false positive rate for different thresholds.

Results: The main observation of this experiment is that the false positive rate is decreasing when the minimum number of required clusters per person is increasing. This result is not very surprising and was to a certain degree expected. Generally, false positive detections usually occur at large distances near the 4 meter range. Around this distance the slicing effect of the distance value starts to appear. At this regions, a wall for example is not only part of one "slice" but in might be distributed over several slices. That means a wall does not exactly look as a flat surface and therefore result in a wrong distribution of surface normals. This happens only for small horizontal clusters with a small width. Large walls are not effected by this. But there are other objects with similar shapes like a human (when considering the segmented clusters). A rounded pillar for example can have a similar normal distribution as a human when considering the segmented clusters in a layer. One solution to decrease the false positive detections would be the consideration of more clusters per person. The graph in figure 7.6 depicts a decreasing false positive error. Since the false positive detections occur mostly in two or three neighboring layers, the rate can be reduced by considering more segments per person. But choosing this threshold is a tradeoff. Standing people can still be detected with a threshold of six or seven clusters, but than the system will not detect sitting person. For the detection of sitting person the visible area of upper body is very important and this consists for a normal sized person of about three to four clusters (with the proposed slice height in Section 7.2.2). In order to still be able to detect sitting persons, a threshold of a least three clusters per person has been chosen. Despite from the described 2D verification, further improvements to reduce the false detections are described in our future outlook in Section 9.

7.2.5 Computation Time

Description: One major aspect related to the processing of large point cloud data is the computation time. Although, the raw input data is reduced to a minimum level in the preprocessing stage, there is still a lot of data left which consumes much of the costly and sparsely available computation time. This is a major problem especially in large robotic systems with e.g. many perception components utilizing 3D data. In this experiment, a robot was navigating through different environments (i.e. apartment, laboratory and corridor) while performing the people detection. During this run, the total processing time of the whole component as well as the processing time of each subtask (preprocessing, segmentation and so on) was recorded. The experiment was

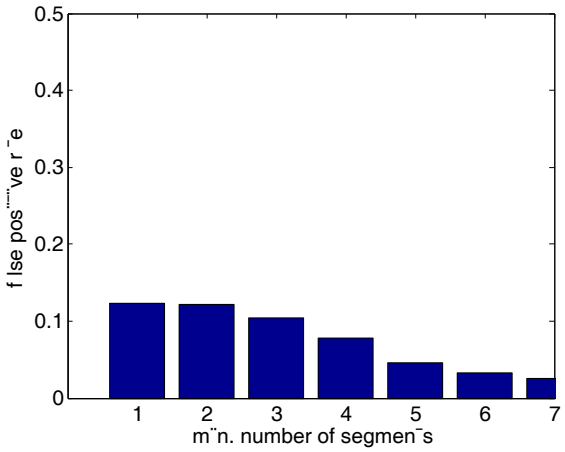


Figure 7.6: False positive detection rates for different qualification thresholds. The threshold describes the required number of minimum positive classified clusters (i.e. as human) per person.

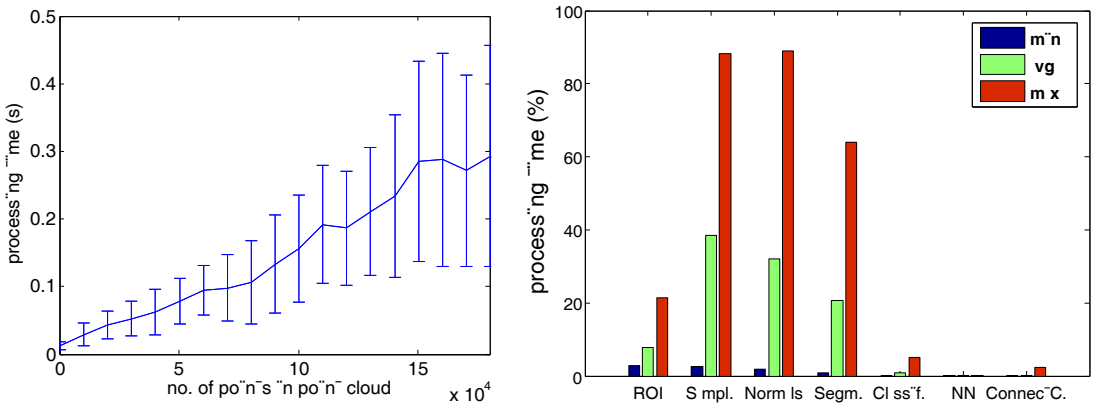
executed on a Intel i7 2.7 GHz with eight cores and 6GB RAM.

Results: The results of the experiments have shown that the component performs the people detection within 91.01 ms in average on the described target hardware. This is an approximate frame rate of 10 Hz. But the actual frame rate is heavily dependent on the number of points in the initial point cloud. Figure 7.7(a) depicts the total processing time relative to the number of points which is left after the ROI building. The time is almost constantly increasing with the total amount of input data which is like expected due to more data that has to be processed by the specific subtasks. An detailed decomposition of the processing time to the corresponding subtask is illustrated in figure 7.7(b). The histogram shows that especially the first subtasks (ROI building, subsampling, normal computation and segmentation) in the processing pipeline consumes the majority of the overall time, because they have to cope with the large data and then provide the reduced data to the remaining subtasks. One solution to speed up things would be a parallelized implementation of the respective subtask. Although GPU-based implementations have been become recently popular (e.g. due to the CUDA framework[14]), it needs a lot of experience to parallelize a give application. A naive and user-friendly parallelization solution has been presented in Section 5.4. The results are presented in the next experiment.

7.2.6 Parallelization

Description: The previous experiment in Section 7.2.5 showed that there are mainly four subtasks which actually consume most of the overall computation time, namely the

[14] **CUDA framework** - `http://developer.nvidia.com/category/zone/cuda-zone`

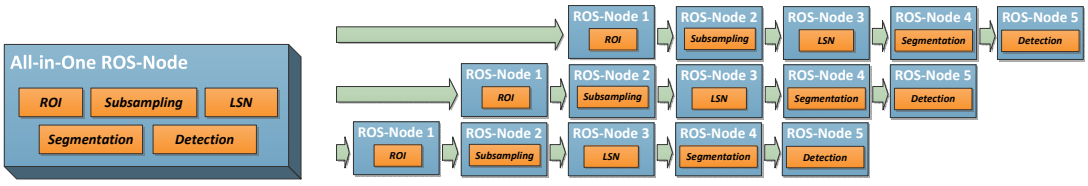


(a) Mean and standard deviation (b) Percentage per subtask

Figure 7.7: A breakdown of the people detection computation time

ROI building, the subsampling, the surface normal computation and the segmentation. In Section 5.4, an outsourcing of these subtasks into separate ROS nodes was proposed to speed up the processing pipeline on multi-core systems. In this experiment, the single node structure was modified to the multi node structure in figure 7.8(b). The navigation path through the different environments from the previous was repeated and the processing time was recorded.

Results: The separation of computational expensive subtasks into separate ROS nodes can result in a speed-up, but only if the processed point cloud consists of a large amount of data. Above 80.000 points, the multi node configuration (MNC) starts to result in a faster computation. While the single node configuration (SNC) consumes significantly more time with an increasing amount of data, the multi node setup almost saturates to a steady timing level. For small data the speed-up of the MNC is very low and is almost rescind by the cost of the data transportation between the nodes. This cost only increases slightly with larger data and is one reason for the advantage of the MNC in comparison to the SNC. During several other experiments, the average number of points in a point cloud have been determined which is about



(a) Sequential pipeline (b) Parallel pipeline

Figure 7.8: Outsourcing of subtasks into separate ROS nodes to achieve a multi stage processing pipeline.

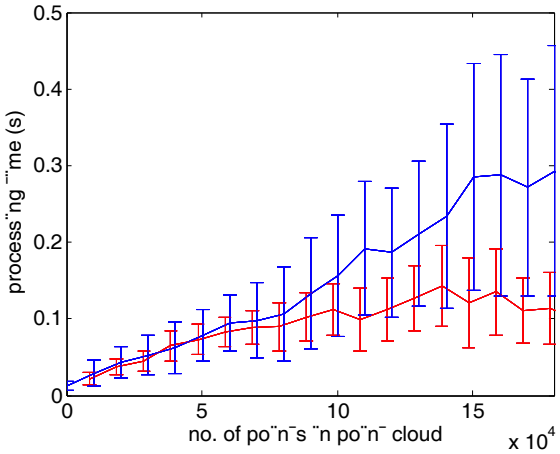


Figure 7.9: Performance comparison of a single node and a multi node implementation

125.386 points after the ROI building. The resulting speed-up for this number of points is

$$S_p = \frac{T_1}{T_p} = \frac{0.182}{0.116} = 1.569$$

where p=5 is the number of nodes of the MNC. This experiment showed that under certain circumstances (i.e. large amount of data), the MNC can results in a speed-up of 1.569 in comparison to the SNC. The splitting of specific subtask into separate nodes can be further an advantage when several components (e.g. 3D object classification) also need some of the preliminary tasks.

7.2.7 Scenario

Description: The final experiment applied a more scenario-like evaluation, where an autonomous mobile service robot tries to find a predefined number of persons in the environment. It is basically derived from the *WhoIsWho* test in the RoboCup@Home rulebook where five people are spread around the apartment (three standing, two sitting). As initial knowledge, the robot has a map of the environment (for navigation purpose) and a set of room poses for each part of the apartment (e.g. kitchen, diner table or living room). When starting the experiment, a script first generates random positions in the map for five persons and also state if they should sit or stand. If the proposed position is blocked, e.g. through a wall or a table, the person should be positioned next to it. If the robot stops navigating and announces that the path is blocked, the person which blocks the path is allowed to go apart. Positions outside the apartment are rejected through a map filter and newly generated. If it is stated that a person should sit and there is actually no opportunity, chair can be put at

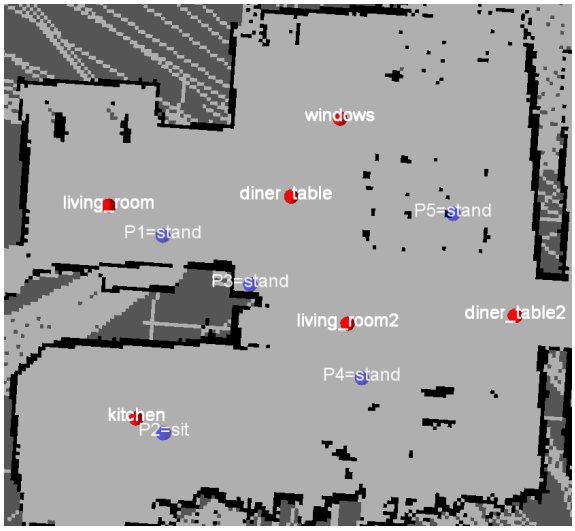


Figure 7.10: Example of generated person positions and their associated state (stand or sit) for the scenario experiment.

the position. The map in figure 7.10 depicts an example of generated person poses. When all persons are placed at the generated positions, the robot generates a random path through all available room poses. The rest of the test consists of executing the *Drive & Search behavior* which is described in Section 6.5. If a person is detected, the robot is going to approach the detected position and announce the height of the detected person.

Results: In total ten runs of the described experiment were executed. The specific setup for each particular run is illustrated in figure B.1 and B.2 in the appendix of this thesis. The figures show the auto-generated positions where the persons had to stand and in which configuration - standing or sitting. Markers illustrate which person have been detected (green circles) and which not (red circles). False positive detections are highlighted with blue circles.

In all cases the robot was able to find at least the two standing persons and always one sitting person. The missed detections where caused by a occlusion through another person or when the person was sitting in a arm chair and only a small part of the shoulder and head was visible. Beside the successful and missing detections, there were quite a lot false positive detections. In each run at least one false positive detection occurred. Due to the fact that a detected person (in this cases a false detection) is approached only once and then stored, the false detections do not effect the overall performance so much. Only the time for approaching the false detection for the first time is gone. But in other scenarios this effect could result in a worse performance. Due to this false detections, the verification in the 2D space

Run	TP standing	TP sitting	FN standing	FN sitting	FP
1	3	2	0	0	2
2	3	1	1	1	2
3	2	2	1	0	1
4	3	2	0	0	2
5	2	1	1	1	2
6	2	2	1	0	1
7	2	1	1	1	1
8	3	2	0	0	2
9	3	1	0	1	2
10	3	1	0	1	1

Table 7.2: Result of 10 executed runs with auto-generated person positions (three standing and two sitting). TP = true positives, FN = false negatives, FP = false positives.

has been proposed in order to reduce them. Other possibilities are discussed in the our future work in Section 9.

Nevertheless, the integration of the people detection component into a higher level behavior was able to successfully detect the majority of people in the environment. Standing people could be detected with a rate of 86.67% and sitting person with 75.00%. Astonishingly, the detection rates from this experiment almost reflect the results acquired in Section 7.2.3.

Chapter 8

CONCLUSION

In this thesis a new concept to detect people in 3D point clouds is proposed, implemented and integrated on a real mobile service robot. Typical use-cases were derived from the recent RoboCup@Home rulebook which describe several requirements for the developed people detection component. The final system fulfills all the requirements discussed in Section 4.2 like person/environment independence, non-static camera and various person poses. One additional requirement was an appropriate sensor type. The recently available Microsoft Kinect sensor was chosen as primary sensor which has become very popular in the field of 3D sensing due to its low price, the respective accuracy and a frame rate of 30 Hz. The camera combines the advantages of LRFs (fast, accurate), monocular cameras (color information) and TOF cameras (3D information).

The preliminary segmentation is based on a top-down/bottom-up technique which yields the capability of detecting partially occluded person, e.g. behind a desk or cupboard. The information gained from the local surface normals enable the system to detect a person in various poses and motions, i.e. sitting on other objects, bended to the front or side, walking fast/slow. As final machine learning technique, a Random Forest classifier is applied which outperformed the opponents AdaBoost and SVM. The presented approach is able to detect people up to a distance of 5 meters with a detection rate of 87.29% for standing and 74.94% for sitting people. The experimental results revealed a certain amount of false positive detections which occured especially at large distances, caused by the discretezation error of Kinect and the resulting inaccurate normal estimation at those distances. A second stage is proposed (but not implemented yet completly) which verfies the detection from the 3D space in a 2D image. The 2D body is segmented into head and shoulder. The parameters of the fitted ellipses and the relation between them build a feature vector for a machine learning classifer. Several measurements during different experiments resulted in a average frame rate of 10 Hz for the detection in the point cloud. This frame rate could be speed up to almost 16 Hz by splitting the computational expensice subtask into separate ROS nodes with less effort and limited experience in parallel computing.

Compared to our previous work [19], the new system benefits from the increased field of view and can handle partially occlusions and sitting people very well. In such cases, our previous laser-based approach definitely resulted in many false negative detections. In comparison to other state-of-the-art approaches, our approach is not restricted to a

static camera setup and is integrated on a real moving service robot. A behavior was implemented which embedded the people detection functionality into a task-based scenario where a fixed amount of sitting and standing person had to be found in an apartment. Several experiments proved the robust detection in real world domestic environment beside some false positive detections. A further contribution of the proposed approach is that the system can detect standing, sitting and partially occluded people with the same trained classifier.

Chapter 9

FUTURE WORK

The experiments exposed that the false detection cause in a major reduction of the detection rate and hence there is a demand for further improvements in that direction. One option - *the verification in 2D space* - is already described in Section 6.4.5. But why do this false positive detections occur? One reason is the increasing discretization error of the Kinect at large distance and the corresponding normal estimation. The other one is the feature descriptor based on this surface normals. When there are objects with similar normal distributions, like e.g. a pillar, they might be detected as a human since they are ranging through several point cloud slices. Concluding this explanation, the current feature vector needs a refinement by adding *additional 2D/3D geometrical and statistical features*.

Another task of the processing pipeline which could be improved/refined is the *segmentation* of a scene. It might be of interest to combine a segmentation in the point cloud with a color-based segmentation in the RGB image and merge the results to a more accurate segmented scene. But the computational complexity should not increase drastically. The segmentation in each space (3D and 2D) can be done in parallel (like the multi node configuration in experiment 7.2.6). And as long the color segmentation will not consume more time than the 3D segmentation only the merging step would increase the overall computation time.

In order to save more computation time, a speed-up in the *normal estimation* would be possible. Instead of searching the k-nearest neighbors in the point cloud, the actual target point can be projected into the image space, taking the neighboring pixels and project them back to the point cloud. We have not done any experiment on this consideration, but it would be a try to see the difference in the computational complexity. Another thought would be the implementation of the complete component on a GPU. Through frameworks like CUDA, the programming of GPU-based software has been eased a lot. Still it needs certain skills and experience to efficiently parallelize a program.

A higher frame rate of the overall system (towards a performance of 30 Hz) would allow a *tracking of detected people in 3D*. The tracking could be performed on the whole body or even on specific body-parts. A separation of the human body according to its skeleton and the corresponding joints could be used e.g. for a gesture recognition system.

A last future improvement concerns the people search behavior described in Section 6.5. The major weakness of the described state machine is the utilization of predefined

room poses. Although they are visited in a random order, it is not considered where people usually occur and where normally not. In order to search for people more efficiently in a environment, an additional people map, based on the navigation map could be established. Each cell of a new people map is initiated with a zero probability. While the robot is performing various other task, the people detection is continuously executed and each time when a person is detected the probability of the cell where the person was detected (and to a certain amount also the neighborhood) is increased. In a scenario where a robot operates as costumer assistant in a supermarket, the robot could explore the environment for e.g. a week continuously and establish in the meanwhile the people map. At the end, the people map reflects place with frequent people detections. When needed, this knowledge can be used to find people more efficient by creating a path through high probability regions. Even after the initial establishment of the map, it can be refinement all the time with new detections.

BIBLIOGRAPHY

[1] Kai Oliver Arras, Oscar Martinez Mozos, and Wolfram Burgard. Using Boosted Features for the Detection of People in 2D Range Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3402–3407, Rome, Italy, 2007. IEEE.

[2] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. H. Matthies. A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle. *The International Journal of Robotics Research*, 28(11-12):1466–1485, July 2009.

[3] Nicola Bellotto and Huosheng Hu. Multisensor-based Human Detection and Tracking for Mobile Service Robots. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(1):167–81, February 2009.

[4] Keni Bernardin, Florian van de Camp, and Rainer Stiefelhagen. Automatic Person Detection and Tracking using Fuzzy Controlled Active Cameras. In *The Seventh IEEE International Workshop on Visual Surveillance (VS2007), IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis; USA, June 2007. Ieee.

[5] Jonathan Boren and Steve Cousins. The SMACH High-Level Executive. *IEEE Robotics and Automation Magazine*, 17(4):18–20, 2010.

[6] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, January 2001.

[7] Thomas Breuer, Geovanny R. Giorgana Macedo, Ronny Hartanto, Nico Hochgeschwender, Dirk Holz, Frederik Hegger, Zha Jin, Christian Atanas Mueller, Jan Paulus, Michael Reckhaus, Jose Antonio Alvarez Ruiz, Paul G. Ploeger, and Gerhard Karl Kraetzschmar. Johnny: An Autonomous Service Robot for Domestic Environments. *Springer Journal of Intelligent and Robotic Systems*, (Special Issue on Domestic Service Robots in the Real World):1–28, 2011.

[8] John F. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–98, June 1986.

[9] Alexander Carballo, Akihisa Ohya, and Shin Yuta. Multiple People Detection from a Mobile Robot using Double Layered Laser Range Finders. In *Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*, number May, Kobe, Japan, 2009.

[10] Baisheng Chen. Indoor and outdoor people detection and shadow suppression by exploiting HSV color information. *Frontiers of Electrical and Electronic Engineering in China*, 3(4):406–410, August 2008.

[11] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790 – 799, 1995.

[12] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, September 1995.

[13] Navneet Dalal Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, San Diego, USA, 2005.

[14] Murat Ekinci and Eyüp Gedikli. Background Estimation Based People Detection and Tracking for Video Surveillance. In Adnan Yazici and Cevat Sener, editors, *Computer and Information Sciences - ISCIS 2003*, volume 2869 of *Lecture Notes in Computer Science*, pages 421–429. Springer Berlin / Heidelberg, Antalya, Turkey, 2003.

[15] Yoav Freund, Robert E Schapire, and Murray Hill. Experiments with a New Boosting Algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy, 1996.

[16] Birgit Graf, Ulrich Reiser, Martin Hägele, Kathrin Mauz, and Peter Klein. Robotic Home Assistant Care-O-bot3 - Product Vision and Innovation Platform. In *IEEE International Conference on Robotics and Automation Society: IEEE Workshop on Advanced Robotics and its Social Impacts - ARSO 2009: Workshop Proceedings*, pages 139–144, Tokyo, Japan, 2009.

[17] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979.

[18] Frederik Hegger. Towards Robust Track and Following of a Human Guide by an Autonomous Mobile Robot. Technical report, Bonn-Rhine-Sieg University of Applied Science, Sankt Augustin, Germany, 2009.

[19] Frederik Hegger. Towards People Detection and Tracking with Two Laser Range Finders. Technical report, Bonn-Rhein-Sieg University of Apllied Science, Sankt Augustin, Germany, 2010.

[20] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-Time Plane Segmentation using RGB-D Cameras. In *Proceedings of the 15th RoboCup International Symposium*, number D, Istanbul, Turkey, 2011.

[21] Chunhua Hu, Xudong Ma, and Xianzhong Dai. A Robust Person Tracking and Following Approach for Mobile Robot. In *International Conference on Mechatronics and Automation*, pages 3571–3576, Harbin, Heilongjiang, China, August 2007. Ieee.

[22] Zha Jin. An Optimized GBNR Sound Localization Algorithm with 4 elements Microphone Array. In *Workshop on Domestic Service Robots in the Real World held at the 2nd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 263–273, Darmstadt, Germany, 2010.

[23] K. Khoshelham. Accuracy Analysis of Kinect Depth Data. In *International Society for Photogrammetry and Remote Sensing (ISPRS)*, Calgary, Canada, 2011.

[24] Margaret A. McDowell, Cheryl D. Fryar, Rosemarie Hirsch, and Cynthia L. Ogden. Anthropometric reference data for children and adults: U.S. population, 1999-2002. *Advance data*, (361):1–5, July 2005.

[25] Niloy J. Mitra and An Nguyen. Estimating Surface Normals in Noisy Point Cloud Data. In *19th ACM Symposium on Computational Geometry*, pages 322–328, San Diego, USA, 2003.

[26] Oscar Martinez Mozos, Ryo Kurazume, and Tsutomu Hasegawa. Multi-Layer People Detection using 2D Range Data. In *Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*, number May, Kobe, Japan, 2009.

[27] Christian Atanas Mueller, Nico Hochgeschwender, and Paul G. Plöger. Towards Robust Object Categorization for Mobile Robots with Combination of Classifiers. In *In Proceedings of the 15th RoboCup International Symposium*, Istanbul, Turkey, 2011.

[28] Rafael Munoz-Salinas, Eugenio Aguirre, Miguel Garcia-Silvente, and Rui Paul. A New Person Tracking Method for Human-Robot Interaction Intended for Mobile Devices. *MICAI 2007: Advances in Artificial Intelligence*, 4827:747–757, 2007.

[29] L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian Detection and Tracking Using Three-dimensional LADAR Data. *The International Journal of Robotics Research*, 29(12):1516–1528, May 2010.

[30] A.S. Ogale and Yiannis Aloimonos. Shape and the stereo correspondence problem. *International Journal of Computer Vision*, 65(3):147–162, December 2005.

[31] Cristiano Premebida and Urbano Nunes. Segmentation and Geometric Primitives Extraction from 2D Laser Range Data for Mobile Robot Applications. In *Robotica 2005 Scientific meeting of the 5th National Robotics Festival*, pages 17–25, Coimbra, Portual, 2005.

[32] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS : an open-source Robot Operating System. In *In Proceedings of the Open-Source Software workshop at the International Conference on Robotics and Automation (ICRA)*, number Figure 1, Kobe; Japan, 2009.

[33] T. Rabbani, F. A. van Den Heuvel, and G. Vosselmann. Segmentation of Point Clouds using Smoothness Constraint. *ISPRS Image Engineering and Vision Metrology*, 1:1–6, 2006.

[34] Yang Ran and Qinfen Zheng. Multi Moving People Detection from Binocular Sequences. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 3:37–40, 2003.

[35] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-Validation. In Ling Liu and M Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US, 2009.

[36] Javier Ruiz-del solar and Jesus Savage. RoboCup@Home Rulebook. Number May, Istanbul, Turkey, 2011.

[37] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität Müchen, 2009.

[38] Radu Bogdan Rusu and Steve Cousins. 3D is here : Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–4, Shanghai, China, 2011.

[39] Junji Satake and Jun Miura. Robust Stereo-Based Person Detection and Tracking for a Person Following Robot. In *Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*, number May, Kobe, Japan, 2009.

[40] Luciano Spinello and Kai Oliver Arras. People Detection in RGB-D Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, USA, 2011.

[41] Luciano Spinello, Kai Oliver Arras, R. Triebel, Roland Siegwart, M. Luber, G.D. Tipaldi, B. Lau, Wolfram Burgard, and Others. A Layered Approach to People Detection in 3D Range Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 55, pages 30–38, Anchorage, Alaska, 2010. IEEE.

[42] Luciano Spinello and Roland Siegwart. Human Detection using Multimodal and Multidimensional Features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3264–3269, Pasadena, USA, 2008.

[43] Luciano Spinello, Roland Siegwart, and Rudolph Triebel. Multimodal People Detection and Tracking in Crowded Scenes. In *In Proc. the AAAI Conf. on Artificial Intelligence: Physically Grounded AI Track (AAAI)*, pages 1409–1414, Chicago, USA, 2008.

[44] Johannes Strom, A. Richardson, and Edwin Olson. Graph-based Segmentation for Colored 3D Laser Point Clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2131–2136, Taipei, Taiwan, 2010. IEEE.

[45] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:I–511–I–518, 2001.

[46] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[47] Sang Min Yoon and Hyunwoo Kim. Real-time multiple people detection using skin color, motion and appearance information. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, pages 331–334, Kurashiki, Japan, 2004. Ieee.

[48] Zoran Zivkovic and Ben Kroese. Part based People Detection using 2D Range Data and Images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 214–219, San Diego, USA, 2007.

Appendix A

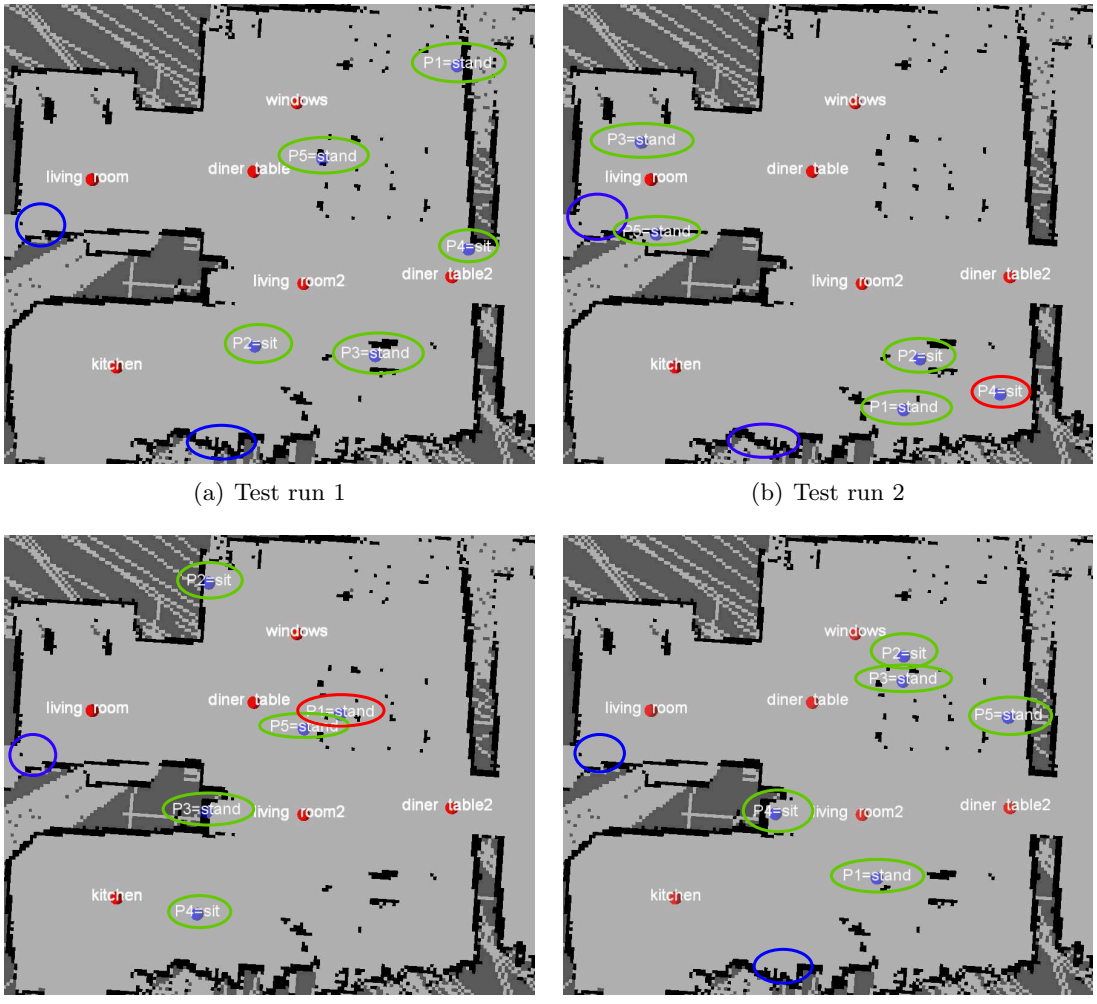
Attached CD-ROM

Attached CD with the following content:

- Thesis as PDF
- BibTex entry as File
- People detection related source code
- Videos of the detection system in action
- Images of the thesis in original size
- References as PDF

Appendix B

Setups and Results of the Scenario Experiment



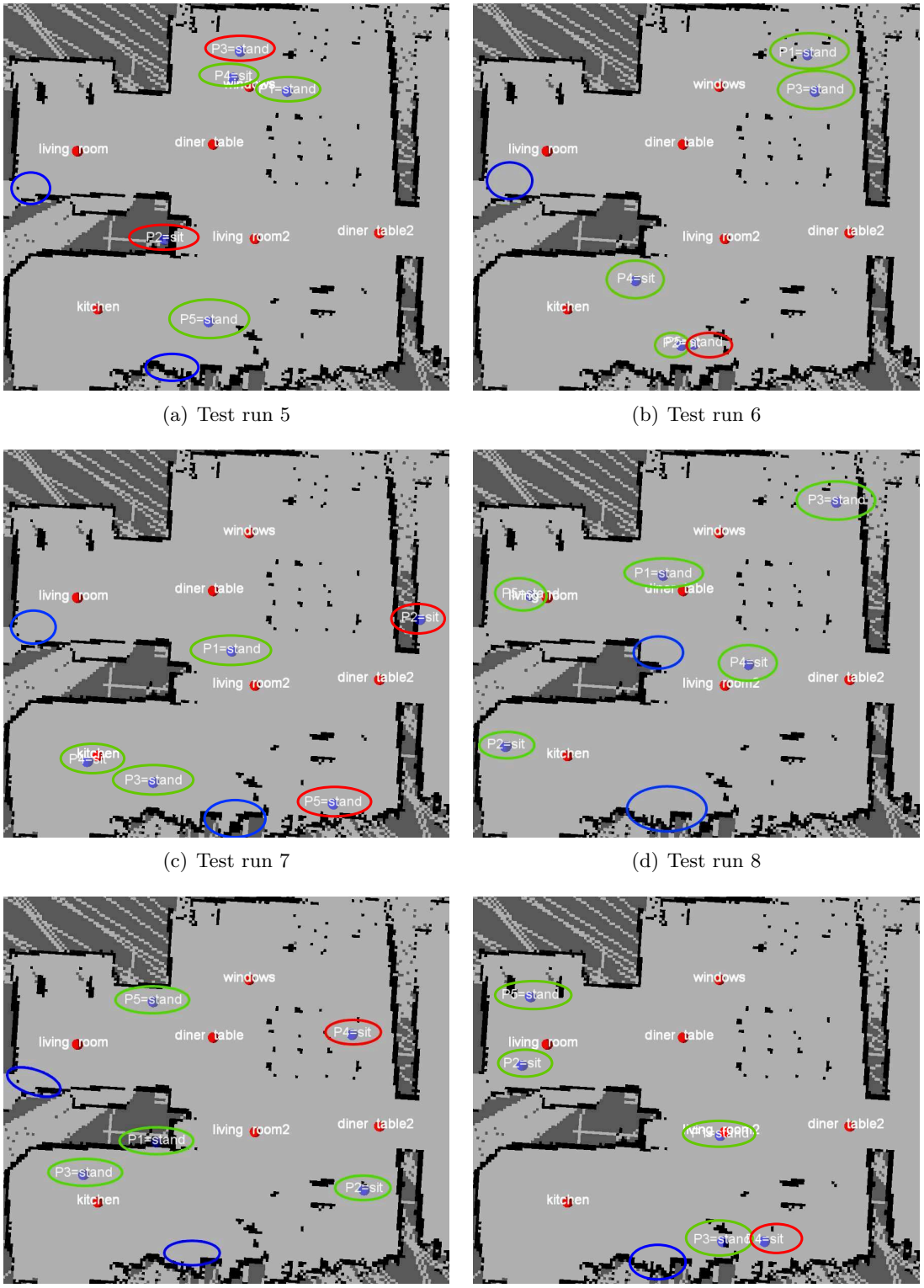
(a) Test run 1

(b) Test run 2

(c) Test run 3

(d) Test run 4

Figure B.1: Setups and results from the test runs 1 - 4 of the scenario experiment where green circles = successful detections, red circles = missed detections, blue = false detections.



(a) Test run 5

(b) Test run 6

(c) Test run 7

(d) Test run 8

(e) Test run 9

(f) Test run 10

Figure B.2: Setups and results from the test runs 5 - 10 of the scenario experiment, where green circles = successful detections, red circles = missed detections, blue = false detections.

Appendix C

Publicly Available Videos

Two videos of the applied people detection approach have been published online at the well-known video platform YouTube:

- Video of one/two person(s) walking randomly, sitting on a chair and performing various body postures. The video has been captured in the RoboCup@Home laboratory at the Bonn-Rhine-Sieg University of Applied Science:

  `http://www.youtube.com/watch?v=d0O4nQE8Qko`

- The second video has been captured in the entrance hall of Bonn-Rhine-Sieg University of Applied Science where many people enter and leave the building. This video shows the performance on many different sized and dressed people. In some cases even a whole group of persons is crossing the FOV where the persons were walking very close to each other:

  `http://www.youtube.com/watch?v=DrJKFQKWFzg`